

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

«На правах рукопису»

УДК 004.01/.08

«До захисту допущено»

Завідувач кафедри

Стіренко С.Г.
(підпис) (ініціали, прізвище)

“ ” _____ 2020 р.

Магістерська дисертація

зі спеціальності: 123. Комп'ютерна інженерія
(код та назва напрямку підготовки або спеціальності)

Спеціалізація: 123. Комп'ютерні системи та мережі

на тему: Метод багатокритеріальної оптимізації для задачі побудови календарного плану виробництва

Виконав (-ла): студент (-ка) _____ курсу, групи _____
(шифр групи)

Гріщенко Костянтин Сергійович
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник проф., д.т.н., проф.Сімоненко В.П.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет (інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра Обчислювальної техніки
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність 123. Комп'ютерна інженерія
(код і назва)

Спеціалізація 123. Комп'ютерні системи та мережі
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Стіренко С.Г.
(підпис) (ініціали, прізвище)

«26» жовтня 2020 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

(прізвище, ім'я, по батькові)

1. Тема дисертації Метод багатокритеріальної оптимізації для задачі побудови календарного плану виробництва

Науковий керівник дисертації проф., д.т.н., проф. Сімоненко В.П.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 26 » жовтня 2020 р. № 3132-с

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження календарне планування виробництва

4. Предмет дослідження алгоритм побудови оптимального календарного плану

5. Перелік завдань, які потрібно розробити:_____

1.Аналіз існуючих алгоритмів побудови розкладів

2.Побудова математичної моделі системи множини робочих центрів з різною продуктивністю, які працюють з перервами

3.Аналіз використовуваних критеріїв оцінки розкладу, побудова агрегованої цільової функції для вирішення задачі оптимізації

4.Розробка та аналіз евристичного алгоритму побудови близького до оптимального розкладу роботи системи з урахуванням заданих критеріїв оптимізації

6. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	проф., д.т.н., проф. Сімоненко В.П.	15.09.2020	09.10.2020
2	проф., д.т.н., проф. Сімоненко В.П.	10.10.2020	27.10.2020
3	проф., д.т.н., проф. Сімоненко В.П.	28.10.2020	01.11.2020

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів дисертації	Примітка
	Вивчення літератури за темою дисертації	10.05.2020	
	Аналіз існуючих методів розв'язання задачі	21.05.2020	
	Розробка математичної моделі	20.10.2020	
	Реалізація алгоритму побудови календарного розкладу	23.10.2020	
	Реалізація системи аналізу отриманого календарного плану	26.10.2020	
	Проведення дослідження якості календарного плану	02.11.2020	
	Оформлення дисертації	14.11.2020	

Студент

Гріщенко К. С.

Науковий керівник дисертації _____
(підпис)

_____ Сімоненко В.П
(ініціали, прізвище)

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Метод багатокритеріальної оптимізації для задачі побудови календарного плану виробництва

студентом: Гріщенко Костянтин Сергійович

Робота складається із вступу та чотирьох розділів. Загальний обсяг роботи: 72 аркуші основного тексту, 15 ілюстрації, 20 таблиць. При підготовці використовувалася література з 46 різних джерела.

Актуальність. Розвиток сучасних систем збуту, розширення спектру пропонованих товарів змушують підприємства до впровадження рішень, які дозволять виробляти продукцію вчасно, відповідно до потреб замовника з найменшими власними затратами.

Для досягнення цих цілей підприємству уже не достатньо просто розширяти виробничі потужності. Потрібно рішення яке дасть конкурентну перевагу без великих капітальних затрат. Календарне планування є саме таким інструментом. Використання автоматизованого розрахунку календарного плану дозволяє досягти економічних показників, які можна порівняти з покупкою нового обладнання. Таким чином в сучасності підприємство не може бути конкурентоспроможним, при відсутності або недостатній якості календарного плану виробництва.

Мета і завдання дослідження. Метою магістерської роботи є збільшення якості календарного плану за рахунок оптимального розподілу операцій між обладнанням і побудови оптимального порядку робіт на обладнанні.

Для досягнення мети дослідження поставлено і вирішено такі завдання:

- аналіз задач управління виробництвом;

- побудова математичної моделі системи з множини робочих центрів з різною продуктивністю, які працюють без перерв;
- аналіз існуючих алгоритмів побудови розкладів;
- аналіз використовуваних критеріїв оцінки розкладу, побудова агрегованої цільової функції для вирішення задачі оптимізації;
- розробка та аналіз евристичного алгоритму побудови близького до оптимального розкладу роботи системи з урахуванням заданих критеріїв оптимізації;
- підтвердження ефективності отриманих результатів при вирішенні практичних завдань промислової розмірності.
- розробка WEB сервісу для розрахунку і аналізу календарних планів.

Об'єкт дослідження – календарне планування виробництва.

Предмет дослідження – алгоритм побудови оптимального календарного плану.

Методи досліджень. В магістерській роботі для досягнення мети і поставлених задач, використано методи теорії розкладів, методи динамічного програмуванні, жадібні алгоритми, методи програмування обмеженнями.

Наукова новизна одержаних результатів роботи полягає у наступному:

- запропоновано двох прохідний алгоритм календарного планування
- розроблено перший прохід алгоритму на основі евристики
- розроблено другий прохід алгоритму на основі програмування обмеженнями
- алгоритми враховують множину критеріїв оптимізації різної ваги
- вага критеріїв може бути налаштована під особливості кожного підприємства

Проведене дослідження дозволяє використання даного алгоритму для розрахунку календарних планів реальних підприємств та дозволяє моделювати зміну продуктивності роботи підприємства від додавання нового обладнання, зміни розкладу роботи, зміни технологічних маршрутів.

Особистий внесок здобувача. В магістерському дослідженні відображено результати самостійно виконаної роботи, в якій застосовано особистий авторський підхід та результати практичних досліджень задачі календарного планування виробництва. Формулювання мети та завдань дослідження проводилось спільно з науковим керівником.

Практична цінність. Отримані результати досліджень дозволяють проводити подальші розробки і дослідження в таких сферах

- вдосконалення методів побудови планів закупки матеріалів;
- оптимізація логістики збуду;
- оптимальні плани обслуговування обладнання;
- перепланування роботи підприємства в режимі online.

Ключові слова

Календарне планування виробництва, теорія розкладів, евристика, програмування обмеженнями, критерії оптимізації календарного плану

ЗМІСТ

ВСТУП.....	Ошибка! Закладка не определена.
РОЗДІЛ 1.....	Ошибка! Закладка не определена.
АНАЛІЗ ЗАДАЧІ ПОБУДОВИ КАЛЕНДАРНОГО ПЛАНУ	Ошибка!
Закладка не определена.	
1.1. Сучасний підхід до управління виробництвом	Ошибка! Закладка не определена.
1.1.1. Загальна структура систем управління виробництвом	Ошибка! Закладка не определена.
1.1.2. Багаторівневий підхід до планування виробництвом	Ошибка! Закладка не определена.
1.1.3. Головний календарний план (MPS)	Ошибка! Закладка не определена.
1.1.4. План потреби в матеріалах (MRP)	Ошибка! Закладка не определена.
1.1.5. Оперативний календарний план (APS)	Ошибка! Закладка не определена.
1.2. Математичні методи вирішення задачі календарного планування виробництва.....	Ошибка! Закладка не определена.
1.2.1. Математична поставка задачі	Ошибка! Закладка не определена.
Висновки до розділу 1	Ошибка! Закладка не определена.
РОЗДІЛ 2.....	Ошибка! Закладка не определена.
РОЗРОБКА АЛГОРИТМУ ПОБУДОВИ КАЛЕНДАРНОГО ПЛАНУ	
.....	Ошибка! Закладка не определена.
2. Огляд методів побудови розкладів	Ошибка! Закладка не определена.

2. Комбінаторна оптимізація **Ошибка! Закладка не определена.**
 - 2.1.1. Класифікація алгоритмів комбінаторної оптимізації **Ошибка! Закладка не определена.**
 - 2.2. Теорія розкладів **Ошибка! Закладка не определена.**
 - 2.3. Жадібний алгоритм..... **Ошибка! Закладка не определена.**
 - 2.4. «Жадібний» рандомізований адаптивний пошук (GRASP) **Ошибка! Закладка не определена.**
 - 2.5. Критерії оптимізації..... **Ошибка! Закладка не определена.**
 - 2.6. Пропонований алгоритм **Ошибка! Закладка не определена.**
 - 2.6.1. Головний цикл евристичного алгоритму пошуку початкового рішення **Ошибка! Закладка не определена.**
 - 2.6.2. Алгоритм вибору інтервалу призначення роботи на обладнання..... **Ошибка! Закладка не определена.**
 - 2.6.3. Алгоритм вибору інтервалу призначення роботи на обладнання..... **Ошибка! Закладка не определена.**
 - 2.6.4. Алгоритм вибору оптимального призначення по множині критеріїв **Ошибка! Закладка не определена.**
 - 2.6.5. Позиційний алгоритм оптимізації..... **Ошибка! Закладка не определена.**
 - 2.6.6. Мінімаксний алгоритм оптимізації..... **Ошибка! Закладка не определена.**
 - 2.7. Алгоритм другого проходу **Ошибка! Закладка не определена.**
 - 2.7.1. Цільова функція алгоритму другого проходу..... **Ошибка! Закладка не определена.**
 - 2.7.2. Обмеження алгоритму другого проходу .. **Ошибка! Закладка не определена.**

2.7.2.1. Обмеження “час переналагодження” ..**Ошибка! Закладка не определена.**

2.7.2.2. Обмеження “календарний час виконання операції”
Ошибка! Закладка не определена.

Висновки до розділу 2 **Ошибка! Закладка не определена.**

РОЗДІЛ 3..... Ошибка! Закладка не определена.

РОЗРОБКА СЕРВІСУ ПОБУДОВИ КАЛЕНДАРНОГО ПЛАНУ

ЗАПРОПОНОВАНИМ АЛГОРИМОМОшибка! Закладка не определена.

3.1. Вимоги до програмного продукту**Ошибка! Закладка не определена.**

3.2. Структура програмного продукту**Ошибка! Закладка не определена.**

3.3. Розробка WEB сервісу..... **Ошибка! Закладка не определена.**

3.4. Розробка інтерфейсу користувача**Ошибка! Закладка не определена.**

3.5. Оцінка розкладу отриманого за допомогою алгоритму**Ошибка! Закладка не определена.**

3.6. Оцінка обчислювальних затрат на розрахунок плану.....**Ошибка! Закладка не определена.**

3.6. Напрямки подальших вдосконалень**Ошибка! Закладка не определена.**

Висновки до розділу 3 **Ошибка! Закладка не определена.**

РОЗДІЛ 4..... Ошибка! Закладка не определена.

РОЗРОБКА СТАРТАП ПРОЕКТУ..... Ошибка! Закладка не определена.

4.1. Ідея проекту **Ошибка! Закладка не определена.**

4.2. Технологічний аудит ідеї проекту **Ошибка! Закладка не определена.**

4.3. Аналіз ринкових можливостей запуску стартап-проекту. **Ошибка! Закладка не определена.**

4.4. Розроблення ринкової стратегії проекту **Ошибка! Закладка не определена.**

4.5. Розроблення маркетингової програми стартап-проекту.... **Ошибка! Закладка не определена.**

Висновки до розділу 4 **Ошибка! Закладка не определена.**

ВИСНОВКИ **Ошибка! Закладка не определена.**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **Ошибка! Закладка не определена.**

ВСТУП

На промислових підприємствах однією з центральних завдань управління є завдання планування виробництва. Виробниче планування і детальне планування завантаження виробничих потужностей дозволяє враховувати динаміку і реальний стан справ у виробничому процесі, щоб формувати календарні графіки відповідно до доступності таких ресурсів, як: обладнання, робоча сила, основні матеріали.

Виробниче планування полягає у визначенні календарних строків виконання певної множини завдань, які, як правило, пов'язані між собою і можуть бути представлені у вигляді структур типу дерево або послідовно-паралельного графу.

Дипломний проект присвячений розробці евристичного алгоритму багатокритеріального APS – планування.

Цілі дипломного проекту: підвищення точності календарного планування дрібносерійного виробництва, в порівнянні з традиційними експертними підходами для промисловості. Для досягнення цілей у роботі вирішуються наступні задачі:

- аналіз задач управління виробництвом;
- побудова математичної моделі системи з множини робочих центрів з різною продуктивністю, які працюють без перерв;
- аналіз існуючих алгоритмів побудови розкладів;
- аналіз використовуваних критеріїв оцінки розкладу, побудова агрегованої цільової функції для вирішення задачі оптимізації;
- розробка та аналіз евристичного алгоритму побудови близького до оптимального розкладу роботи системи з урахуванням заданих критеріїв оптимізації;
- підтвердження ефективності отриманих результатів при вирішенні практичних завдань промислової розмірності.
- розробка WEB сервісу для розрахунку і аналізу календарних планів.

РОЗДІЛ 1

АНАЛІЗ ЗАДАЧІ ПОБУДОВИ КАЛЕНДАРНОГО ПЛАНУ

1.1. Сучасний підхід до управління виробництвом

1.1.1. Загальна структура систем управління виробництвом

Сучасний розвиток технологій дозволяє реалізовувати системи управління виробництвом в режимі on-line. Така швидкість реакції на всіх рівнях системи дозволяє досягти надзвичайного росту ефективності підприємства, порівняного з ростом від капіталовкладень в нове обладнання і виробничі підрозділи. Сучасна управління виробництвом являє собою замкнуте коло: основним інструментом управління є оперативний календарний план. Календарний план розділяється на частини(виробниче завдання), кожна відповідає своєму екземпляру обладнання. Виробничі завдання відправляються працівникам закріпленим за відповідним обладнанням на мобільний або WEB додаток. Додаток дозволяє забезпечити чітке слідування виконавцем плану розрахованого системою, так як виконавець бачить яку операцію йому потрібно виконувати в даний час і на якому обладнанні. Додаток дозволяє вести збір даних про фактичне виконання плану: початок роботи, завершення роботи, реєстрацію інцидентів : поломка обладнання, відсутність необхідного для виконання роботи матеріалу. Додатково для даних цілей можуть використовуватись пристрої IoT. Отриманий зворотній зв'язок аналізується системами диспетчеризації і обліку, які ініціює перерахунок календарного плану з врахуванням отриманих даних. Прикладам реакції на зворотній зв'язок є перерозподіл робіт з зламаного обладнання на альтернативне, вилучення з плану фактично виконаних робіт. Принципова схема управління виробництвом представлена нижче (рисунок 1.1). Виділено наступні завдання: планування, організація виробництва, виробництво, облік та оперативний перерахунок календарного плану.

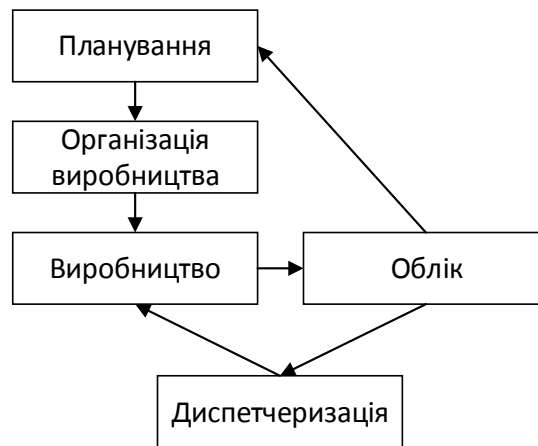


Рисунок 1.1 – Принципова схема управління виробництвом

1.1.2. Багаторівневий підхід до планування виробництвом

Концепція багаторівневого планування дозволяє надати лише необхідні дані для кожної служби підприємства. Технічно це дозволяє вибрати алгоритми, які найбільше підходять для заданої точності й горизонту плану.

В розрізі даної концепції основними рівнями виробничого планування є:

- Перспективний планування продажів і операційної діяльності (Sales and Operations Planning - SOP);
- Головного календарний план виробництва (Master Production Scheduling - MPS);
- Об'ємно - календарне планування потреби в матеріалах (Material Requirement Planning - MRP);
- Об'ємно - календарне планування фонду і завантаження виробничих потужностей (Capacity Requirement Planning - CRP);
- Оперативне календарно - оптимізаційне формування розкладу виробництва з урахуванням обмежень (Advanced Planning System - APS);
- Система контролю виконання виробництва (Manufacture Execution System – MES).

Таблиця 1.1 – Характеристика рівнів планування

Рівень	Об'єкт планування	Горизонт	Інтервал
SOP	Група товарів	рік	місяць
MPS	Замовлення	квартал	доба
MRP	Об'єм затребуваних деталей, покупних матеріалів й комплектуючих під кожне замовлення	місяць	доба
CRP	Завантаження обладнання, трудових ресурсів	місяць	тиждень
APS	Виконання операцій на обладнанні	тиждень	хвилина

1.1.3. Головний календарний план (MPS)

Головний календарний план виробництва (MPS) має лінійну структуру. MPS містить отримані замовлення замовників, заявки на поповнення складу. MPS формується на підставі прогнозу продажів, замовлень покупців, необхідних розмірах страхових запасів і т.д. MPS-план є ковзаючим планом, в який нові замовлення включаються в міру їх надходження (на практиці, зазвичай, один- два рази на тиждень або щодня) та планування виконується від поточної дати до дати випуску всіх прийнятих у виробництво замовлень. Ковзаюче планування, в порівнянні з традиційним для вітчизняної промисловості місячним плануванням, дозволяє істотно підвищити оперативність реагування виробництва на потреби клієнтів і забезпечити більш рівномірне завантаження потужностей.

1.1.4. План потреби в матеріалах (MRP)

На підставі MPS-плану і технології виробництва виконується розрахунок деталізованих оперативних виробничих планів. MRP план має структуру дерева. Вузлами є технологічні операції. Вузол містить планову кількість деталей для проходження операції для даного замовлення і фактично оброблену кількість деталей на даній операції.

Фактично виконана кількість розраховується відповідно до інформації отриманій з системи обліку. Таким чином MRP містить актуальний об'єм робіт для подальшого планування на обладнання.

MRP-план по позиції плану випуска. Режим групування: По вузлам

	Наименование	Техмаршрут	№ оп.	Операция	ЕИ	План	Факт	Изг.	НЗП	РЦ	Изготовитель	Получатель	Заказ
2167.000	Регулятор частоты вращения РО	2167.000 РЦ			шт	2					УР (427)	УР (427)	
	Кольцо 005-007-14-2-043 ОСТ 1 00980-80				шт	2					АО "ФЭД" (0)	УИ (422)	
	Кольцо 17-1,2-1 ОСТ 1 10788-85				шт	2					АО "ФЭД" (0)	УИ (422)	
	1850.010-01 (061) Уплотнение				шт	2					АО "ФЭД" (0)	УИ (422)	
	1850.020-01 Обойма				шт	2					АО "ФЭД" (0)	АО "ФЭД" (0)	
	1850.001-01 Обойма				шт	2					АО "ФЭД" (0)	АО "ФЭД" (0)	
	Круг В1-28 ГОСТ 2590-2006/ 12Х18Н10Т				кг	0.15					АО "ФЭД" (0)	АО "ФЭД" (0)	
	1850.006-01 Шпонка				шт	4					АО "ФЭД" (0)	АО "ФЭД" (0)	
	1850.002-01 Кольцо				шт	2					АО "ФЭД" (0)	АО "ФЭД" (0)	
	1850.003-01 Пружина				шт	2					АО "ФЭД" (0)	АО "ФЭД" (0)	
	1850.004-01 Шайба				шт	2					АО "ФЭД" (0)	АО "ФЭД" (0)	
	1850.005-01 Шайба				шт	2					АО "ФЭД" (0)	АО "ФЭД" (0)	
	1850.007-01 Кольцо				шт	2					АО "ФЭД" (0)	АО "ФЭД" (0)	
	Кольцо 015-018-19-2-061 ОСТ 1 00980-80				шт	2					АО "ФЭД" (0)	АО "ФЭД" (0)	
	1850.010-03 (061) Уплотнение				шт	2					АО "ФЭД" (0)	УИ (422)	
	1850.020-03 Обойма				шт	2					АО "ФЭД" (0)	АО "ФЭД" (0)	

Рисунок 1.2 – Дерево MRP-плану виробництва

1.1.5. Оперативний календарний план (APS)

Завданням оперативного календарного оптимізаційного планування виробництва (APS - планування) є виконання MPS- та/або MRP- планів виробництва оптимальним чином за допомогою оптимізації розкладу роботи обладнання.

У результаті оптимізації досягається виконання виробничих замовлень у строк або з мінімальними порушеннями, більш щільне завантаження потужностей, зменшення тривалості та кількості переналагоджень

устаткування, скорочення часу пролежування та обсягів незавершеного виробництва, скорочення загальної тривалості виробничого циклу і, таким чином, збільшується пропускна здатність підприємства і прискорюється швидкість обороту матеріальних і фінансових ресурсів.

APS-план формується з урахуванням обмеженого завантаження потужностей та інших виробничих ресурсів. У результаті виходять деталізовані позмінні поопераційні виробничі плани для кожного робочого центру з ефективною черговістю виконання замовлень. APS/MES-планування є розвитком об'ємно-календарного CRP-планування потужностей з обмеженим завантаженням. Принциповими відмінностями є:

- відсутність інтервалів планування, які плануються об'ємним способом;
- розташування всіх робіт на часовій осі з точністю до хвилин.

Як наслідок з попереднього пункту, є можливість обліку при плануванні реальної внутрішньозмінної черговості робіт, часу переналагоджень, часу міжопераційного пролежування і переміщення між робочими центрами. Формування плану виконується по одному або декільком критеріям оптимізації.

1.2. Математичні методи вирішення задачі календарного планування виробництва

Математичною основою APS -планування є теорія розкладів – спеціальний розділ математичного програмування і дослідження операцій, який займається завданнями побудови оптимальних послідовностей робіт (розкладів) на виділених для цього робочих центрах з урахуванням різних обмежень.

У зв'язку з тим, що універсальні алгоритми математичної оптимізації є не поліноміально вирішуваними, так званими NP-важкими з обчислювальної точки зору, то математичною основою календарного планування в проекті є комбіноване використання різних евристичних і комбінаторних алгоритмів теорії розкладів.

«Часовий» характер задач теорії розкладів виділяє їх в особливий клас, що істотно відрізняється від «об'ємних» економічних завдань. Якщо в останніх потрібно відповісти на запитання, що і скільки виробляти, то в задачах ТР необхідно визначити, коли, в якій послідовності виконувати роботи. Ця різниця в суті завдань визначає відмінність в методах та можливості їх вирішення. Для задач об'ємного характеру розвинений досить потужний апарат, головним чином математичного програмування, що дозволяє, загалом, з успіхом добиватися їх вирішення. Для завдань ТР вирішальний апарат розвинений в набагато меншому ступені.

Пошук оптимального або близького до оптимального розкладу здійснюється за допомогою одного з 4 підходів:

- математичного програмування;
- комбінаторного підходу;
- евристичного підходу;
- статистичного (ймовірнісного) підходу.

Багато з евристичних, ймовірнісного або чисельних методів у теорії розкладів передбачають необхідність складання досить великого числа розкладів для кожного конкретного завдання. Для опису цих методів крім типів розкладів, введених вище, корисно ввести декілька типів алгоритмів складання розкладів, так як існує досить багато істотно різних способів отримання компактних розкладів.

У всіх алгоритмах складання розкладів використовується множина, що складається з $G = \sum_{i=1}^n g_i$ операцій, причому на кожному кроці вибирається одна з них і їй приписується момент початку виконання. Порядок вибору та метод призначення моменту початку операції і визначають характер алгоритму.

Найбільш важлива відмінність існує між однократними і коректованими алгоритмами. Перші характеризуються тим, що обраний момент початку кожної операції залишається незмінним в ході подальшого складання розкладу. При цьому, однак, потрібно, щоб правила вибору чергової операції і

призначення моменту початку її виконання визначалися б суворої послідовністю в безлічі операцій i , отже, дозволяли б зробити рівно G призначень моментів початку виконання. Для коректованого алгоритму істотно, що кожен з уже призначених моментів початку виконання може бути змінений в ході подальшого складання розкладу. Майже всі алгоритми, для яких складено програми на ЕОМ, є однократними.

У данному проекті розглядається розробка програмного забезпечення для однократних алгоритмів.

1.2.1. Математична поставка задачі

Маємо систему за n продуктів і технологічний маршрут виготовлення кожної продукції j складається з довільної кількості операцій O_{1j}, \dots, O_{n_jj} .

Кожна операція характеризується наступними властивостями:

- d_{ij} – директивний строк виконання;
- u_{ij} – пріоритет операції i для виготовлення продукції j ;
- w_i – вага критерія i ;
- k_{ij} – значення критерія 1, при умові, що наступною буде призначена робота ij ;
- E – порядок виконання операцій;
- нормативи тривалості виконання операції на допустимому робочому центрі. Кожен норматив характеризується номером робочого центру k та t_{ijk} .

Нехай маємо m робочих центрів $M_1, M_2, \dots, M_k, M_m$. Кожний робочий центр характеризується наступними властивостями:

- k – номер робочого центру;
- W_k – множина допустимих операцій для виконання на робочому центрі k ;

- R_k – множина вікон у розкладі робочого центра. Кожне вікно характеризується номером вікна l , часом початку $t_{\text{поч}}$ і закінчення $t_{\text{зак}}$ вікна, ознакою зайнятості вікна та посиланням на виконувану операцію O_{ij} ;
- тривалості переналагоджень між допустимими операціями задаються матрицею переналагоджень:

$$\begin{array}{cccc}
 & A & B & \dots \\
 A & 0 & \tau_{AB} & \dots \\
 B & \tau_{BA} & 0 & \dots \\
 \dots & \dots & \dots & 0
 \end{array}$$

де $A, B, \dots \in W_k$

- тривалості переміщень деталей між робочими центрами задаються матрицею переміщень:

$$\begin{array}{cccc}
 & A & B & \dots \\
 A & 0 & \tau_{AB} & \dots \\
 B & \tau_{BA} & 0 & \dots \\
 \dots & \dots & \dots & 0
 \end{array}$$

де $A, B, \dots \in M$

Цільова функція

Розглянемо найпопулярнішу на практиці цільову функцію мінімізації загальної тривалості виконання операцій у розкладі з врахуванням директивних строків та пріоритетів виготовлення продукції.

Нехай $G = f(O, M, E)$ – сформований розклад. Тоді цільова функція має вигляд:

$$T = f(G) \rightarrow \min$$

Обмеження

Обмеження на порядок виконання операцій такі:

- між усіма операціями технологічного процесу виробництва однієї продукції задано E – суворе відношення передування $O_{1j} \rightarrow O_{2j} \rightarrow \dots \rightarrow O_{n,j}$ (маршрут обробки деталі);

- директивний строк виконання операцій не спадаючий: $d[1] \leq \dots \leq d[n]$;
- пріоритет виконання серед операцій з одним директивним строком не спадаючий: $u[1] \leq \dots \leq u[n]$.

Обмеження на функціонування робочих центрів наступне: кожен робочий центр виконує одночасно не більше однієї операції.

Висновки до розділу 1

В даному розділі була сформульована змістовна та математична постановка задачі складання APS -плану виробництва з урахуванням обмежень.

Задачу APS -планування можна віднести до класу задач теорії розкладів. З урахуванням того, що розмірність промислових даних не дозволяє використовувати точні методи знаходження розв'язку, був розроблений та обґрунтований евристичний алгоритм формування розкладу виконання операцій, який, у більшості випадків, призводить до ефективного розв'язку за прийнятний час.

РОЗДІЛ 2

РОЗРОБКА АЛГОРИТМУ ПОБУДОВИ КАЛЕНДАРНОГО ПЛАНУ

2. Огляд методів побудови розкладів

Завдання побудови розкладу є завданням оптимізації. Для вирішення завдань оптимізації існують неперервні і дискретні методи оптимізації.

Існують точні і наближені методи безперервної оптимізації. Практично всі точні методи вимагають опуклості області визначення задачі. Завдання побудови розкладу в загальному вигляді, а саме такими бувають практичні завдання, цій умові не задовольняє. Тому, на практиці, для застосування залишаються тільки наближені методи, які вимогливі по обчислювальних ресурсів і не гарантують отримання результату, а одержувані практичні рішення практично важко застосувати. Практичні дослідження застосовності багатьох методів неперервної оптимізації показали, що для завдання з 2-ох виробів, 10-и робіт, 5 робочих центрів в умовах безперервного часу (практичні завдання такими не бувають) за кілька хвилин роботи алгоритму не завжди виходить не тільки оптимальне рішення, але навіть і допустиме. За цей час описану задачу людина вирішує оптимально вручну. Тому використовувати такі методи не доцільно з практичної точки зору.

Завдання побудови розкладу можна сформулювати як завдання дискретної оптимізації - це завдання знаходження екстремуму функції, заданої на дискретній (найчастіше кінцевій) безлічі точок. Методи дискретної оптимізації вимагають менше обчислювальних ресурсів, а отже, завдання можна вирішити швидше і отримати кращий результат.

З найбільш відомих точних методів дискретної оптимізації можна навести наступні:

- повний перебір;
- метод гілок і меж;
- динамічне програмування.

Але для практичних завдань розмірність завдань, обчислювальна складність цих методів і можливості сучасної комп'ютерної техніки не дозволяють застосувати практично.

Завдання побудови розкладу можна сформулювати, як задачу комбінаторної оптимізації. Саме в «комбінаторної оптимізації» проведено безліч досліджень і отримано багато теоретичних результатів для вирішення завдань побудови розкладів. Тому в подальшому у даній дисертації зупинимося на методах комбінаторної оптимізації.

2. Комбінаторна оптимізація

Комбінаторна оптимізація (КО) - область теорії оптимізації в прикладній математиці, пов'язана з дослідженням операцій, теорією алгоритмів і теорією обчислювальної складності. Комбінаторна оптимізація полягає в пошуку оптимального об'єкта в кінцевій безлічі об'єктів, чим дуже схожа на дискретне програмування. Методи комбінаторної оптимізації прийнято ділити на точні і наближені. Завдання побудови розкладу з урахуванням практичних вимог, є NP-складною, згідно з визначенням, тому що науці поки що невідомий поліноміальний алгоритм її рішення, але й відсутність його так само не доведено. Для NP-складних задач існує єдиний точний метод знаходження розв'язку - це повний перебір. Обчислювальна складність повного перебору порядку $n!$, де n - це число, що дорівнює кількості різних можливих призначень «Робота - робочий центр». Іноді, завдання побудови розкладу можна вирішити методом гілок і меж (МВГ). Обчислювальна складність МВГ лежить в межах від n^2 до $n!$ Для практичних задач, у яких тисячі робіт і сотні робочих центрів - це приблизно така ж обчислювальна складність. Сучасна обчислювальна техніка не справляється з такими розмірностями. Тому, практично, сьогодні можна застосовувати тільки наближені методи КО.

2.1.1. Класифікація алгоритмів комбінаторної оптимізації

При класифікації алгоритмів КО насамперед слід розділяти точні і наближені методи. Перші володіють властивістю гарантовано знаходити оптимальне рішення. Однак, майже всі практично важливі завдання КО є NP-повними: для них невідомо жодного точного алгоритму з поліноміальною складністю. Крім того, постановки завдань часто бувають з даними, які мають певні похибки, що остаточно робить значні обчислювальні витрати для знаходження точного рішення невиправданими. На противагу точним методам, сучасні наближені алгоритми комбінаторної оптимізації дають змогу одержувати рішення "високої якості" за прийнятний, з практичної точки зору, час. За цією, а також з інших причин, все більше уваги в науковій літературі приділяється наближеним методам комбінаторної оптимізації. За останні десятиліття запропоновані десятки різних підходів, що володіють тими чи іншими властивостями і характеристиками. Для виділення характерних відмінних рис існуючих наближених методів комбінаторної оптимізації запропонована наступна класифікація (див. Таблицю 3.1).

Таблиця 3.1. Класифікація наближених алгоритмів комбінаторної оптимізації.

Класи	Приклади
По принципу прийняття рішення	
Детерміновані	Локальний пошук
Недетерміновані (стохастичні)	Імітаційний отжиг
По структурі	
Простий алгоритм	Локальний пошук
Гібридний алгоритм	
Метаевристика	Метод мурашиних колоній, меметичний алгоритм
Гібридна метаевристика	

По типу використовуваних просторів	
Послідовні (конструктивні)	Евристика «йди до найближчого»
Траекторні	Локальний пошук
Ітераційні	G-алгоритм [Гуляницький, 2006]
Популяційні	Метод мурашиних колоній, Н-метод [Гуляницький, 2008]
За кількістю використовуваних систем околиць	
Одна	Імітаційний отжиг
Безліч	Пошук в змінних околицях
По типу цільової функції	
Статична	Метод мурашиних колоній, генетичний алгоритм
Динамічна	Керований локальний пошук
За типом траєкторії (для ітераційних методів)	
Траекторного-неперервні	G-алгоритм, табу-пошук
Траекторного-розривні	GRASP
За використанням пам'яті	
Без використання пам'яті	Локальний пошук, G-алгоритм, стандартний генетичний алгоритм
З короткочасною пам'яттю	Простий табу-пошук
З довготривалою пам'яттю	Динамічний локальний пошук [21]

З обома видами пам'яті	Табу-пошук з використанням критеріїв прагнення
За оцінкою точності	
З апостеріорної оцінкою точності	Локальний пошук з відомою нижньою межею
З апріорної оцінкою точності	ϵ -наближені алгоритми
Збіжність за рішенням	
Збіжні з деякою вирогідністю	Мурашині алгоритми класу τ min ACO [12]
Не збіжні	Стандартный генетический алгоритм
Без оцінки збіжності	Н-метод
Збіжність за значенням	
Збіжні з певною ймовірністю	Мурашині алгоритми класу τ min ACO [12], G-алгоритм
Не збіжні	Стандартний генетичний алгоритм
Без оцінки збіжності	Н-метод

Для пояснення класифікації розглянемо такі визначення. Збіжністю за рішенням називається досягнення з певною ймовірністю алгоритмом стану, коли одне і те ж оптимальне рішення буде генеруватися постійно;

Збіжністю за значенням називається досягнення з певною ймовірністю алгоритмом стану, коли алгоритм згенерує довільне оптимальне рішення хоча б один раз.

На практиці цікавить отримання рішення задачі КО за відведений на побудову розкладу час. Тому застосовувані алгоритми повинні давати результат за короткий час.

Також, на практиці цікавить отримання рішення задачі КО, в нашому випадку - отримання розкладу, оптимізованого по заданому критерію. Тому більш важливі для вибору алгоритму оптимізації останній клас характеристик алгоритму - збіжність за значенням. Решта класів характеристик не так важливі для оптимізації рішення. Якщо дивитися глибше, то вони більше характеризують структуру і поведінку алгоритму, ніж отримання оптимального рішення.

Збіжність за рішенням для практики має невеликий сенс, тому що отримати оптимальне рішення задачі великої розмірності за відведений на планування короткий час практично неможливо.

2.2. Теорія розкладів

Теорія розкладів - розділ дискретної математики, що займається проблемами впорядкування. Перед нею ставиться таке завдання дискретної оптимізації: побудувати розклад, який мінімізує час виконання робіт, вартість робіт, довжину розкладу, мінімальні відхилення від термінів та інші. Розклад - це вказівка, на яких машинах і в який час повинні виконуватися роботи.

У теорії розкладів були побудовані алгоритми, які вирішують поставлену задачу за час $O(n * \log n)$, але для одного або двох робочих центрів.

Для реальних промислових підприємств застосування напрацювань теорії розкладів свого часу дало певний прорив. Це пов'язано з тим, що застосовуючи сортування для робіт обчислювальну складність завдання можна істотно знизити до $O(n * \log n)$ і побудувати «однопрохідні» алгоритми формування розкладів. Зазначені алгоритми не будували «математично» оптимальний розклад, але упорядкування робіт, такі, як LPT, FPT, CR та інші дозволили отримати практичні рішення для великої кількості робіт.

Якщо дивитися на підхід з точки зору класифікації методів КО, то теорія розкладів використовувала упорядкування, як «евристики», а всередині

алгоритму неявно використовувався «жадібний» принцип, який по-іншому можна назвати: «йди до ближнього», або «локальний пошук» .

Недоліком методів теорії розкладів є те, що досліджені вченими сортування використовували найчастіше, один критерій оптимізації (дуже рідко - два), що для практичних завдань управління виробництвом недостатньо. Комплексні критерії складно декомпонувати, а реалізація практичних бізнес-критеріїв часто не підпадала під відомі теореми. На практиці необхідно не тільки витримати терміни замовлень, що в принципі не завжди і можливо, але при цьому мінімізувати всі непродуктивні витрати, які тільки можна порахувати (налагодження, пролежування в НЗВ, додаткові витрати на оплату праці та інших ресурсів, зменшити час проходження замовлення і т.д.)

Тому область застосування напрацювань теорії розкладів можна описати правилом: застосовувати, якщо немає часу. Тобто для великих розмірностей, при невеликому виділеному часі на розрахунок. Для невеликих розмірностей, коли є час, потрібно застосовувати спрямовані перебори більшої кількості можливих варіантів, і відбирати кращі за необхідними бізнес-критеріями.

2.3. Жадібний алгоритм

Жадібний алгоритм (англ. Greedy algorithm) - алгоритм, що полягає в прийнятті локально оптимальних рішень на кожному етапі, допускаючи, що кінцеве рішення також виявиться оптимальним.

Іноді, такий підхід призводить до глобального оптимуму. Наприклад, якщо структура завдання задається матроїдом, тоді застосування жадібного алгоритму видасть глобальний оптимум.

Приклад

Відвідувач конференції хоче відвідати найбільшу кількість доповідей, які походять в різних секціях. Відомо часи початку S_i закінчення F_i кожної доповіді. Визначити, яку максимальну кількість доповідей може відвідати відвідувач.

Рішення «жадібним» алгоритмом:

Кількість доповідей $K := 0$

$T :=$ час початку конференції

Цикл (поки є непрострочені доповіді (існують $S_i > T$):

- сортуємо залишилися непрострочені доповіді в порядку зростання часу закінчення F_i , якщо часи закінчення збігаються, то за мінімальним часом початку S_i ;
- вибираємо перший з сортування номер доповіді, запам'ятовуємо його номер n ;
- збільшуємо лічильник доповідей $K := K + 1$;
- запам'ятовуємо час закінчення $T := F_n$;

кінець циклу.

Висновок: K

Обчислювальна складність такого алгоритму $O(n^2)$. Можна довести, що для конкретного прикладу обчислювальна складність $O(n * \log n)$, тому що заздалегідь можна впорядкувати доповіді, обчислювальна складність упорядкування $n * \log n$. Іншими словами, задачу можна звести до методів рішення теорії розкладів.

2.4. «Жадібний» рандомізований адаптивний пошук (GRASP)

Абревіатуру GRASP (Greedy Randomized Adaptive Search Procedure) розшифровують, як процедуру жадібного адаптивного рандомізованого пошуку. Алгоритмічно, на кожному кроці «жадібної» евристики часткові рішення оцінюються, а кращі зберігаються в спеціальній обмеженій множині S . Результатом ітерації є випадковий вибір рішення з множини S . Останньою дією кожної ітерації є адаптація «жадібної» функції з урахуванням вже отриманих раніше рішень.

Хороші алгоритми жадібного адаптивного пошуку використовують жадібні евристики з вдалою рандомізацією. Але краще, якщо при цьому вважається не одне, а кілька розкладів паралельно.

Якщо аналізувати напрацювання теорії розкладів, то варто відзначити аналогію з теоремою Монте-Карло, яка використовувала на кожній ітерації випадковий вибір роботи для призначення на верстат. Відомо, що деякі класи «мурашиних» алгоритмів сходяться за ймовірністю до оптимуму. Якщо розглянути цей алгоритм більш детально, то функція переваги є гібридною метаявристикою (поєднанням евристики і ймовірності) і при цьому розраховується не один розклад, а кілька паралельно (кількість розкладів = розмір популяції). Тому реалізацію практичних алгоритмів оптимізації розкладу необхідно виконувати в цьому напрямку.

2.5. Критерії оптимізації

Підхід до критеріїв оптимізації у даній дисертації спрямований на вирішення питань бізнесу. Для бізнесу важливо максимізувати прибуток (іншими словами віддачу від бізнесу), а також закласти базис для успішності бізнесу в майбутньому. Оскільки система орієнтована на побудову виробничих розкладів, то завдання бізнесу транслуються на плановий відділ таким чином:

- необхідно витримувати об'ємні показники плану, необхідні для бізнесу;
- необхідно виконувати замовлення в заявлений термін;
- при виконанні замовлень зменшити власні витрати;
- якість продукції повинна відповідати заявленим вимогам.

Акцент на об'ємні показники дуже важливий, тому що вони гарантують допустиму частку постійних витрат в собівартості продукції. Таким чином, продукція стає рентабельною, а бізнес прибутковим.

Виконання замовлень в строк характеризує виробника в очах клієнта, як надійного постачальника продукції.

Якість продукції - характеристика можливості застосовувати клієнтом продукцію за призначенням.

Виконання всіх цих вимог - запорука отримання нових замовлень в майбутньому.

Декларація тривалих термінів і високих цін дозволяє збільшити рівень обслуговування замовлень і підтримувати рентабельність продукції, але в перспективі відторгає клієнта від виробника. Якщо в конкурентів можна отримати замовлення швидше і дешевше, або якісніше, то клієнт йде. Тому важливо витримувати баланс цін і термінів з пропозиціями конкурентів на ринку.

Для того, щоб підприємство виробляло якісну продукцію система формує розклад відповідно до закладених вимогами технології: де робити, скільки робити, з чого і т.д.

Мінімізація власних витрат є важливим фактором, тому що дозволяє збільшувати прибуток в умовах обмежень на відпускні ціни.

Крім цього, на практиці зібрати за короткий час всю нормативно-довідкову інформацію, необхідну для реалізації єдиного формалізованого комплексного вирішення неможливо. Такі бази знань і формалізовані критерії створюються роками. Тому для можливості практичного використання застосовується метод багатокритеріальної оптимізації та управління вагами критеріїв користувачем.

Для управління виробництвом використовуються такі об'єкти управління:

- замовлення;
- роботи;
- робочі центри;
- персонал.

Тому критерії оптимізації націлені на управління цими об'єктами. Окремо виділяється область комплексних критеріїв, які одночасно керують декількома об'єктами управління.

Для управління виробництвом використовуються такі об'єкти управління:

- замовлення;
- роботи;
- робочі центри;
- персонал.

Тому критерії оптимізації націлені на управління цими об'єктами. Окремо виділяється область комплексних критеріїв, які одночасно керують декількома об'єктами управління.

Таблиця 3.2 – Критерії оптимізації

Об'єкт управління	Критерії оптимізації
Замовлення	Мінімізувати зриви термінів випуску замовлень
	Мінімізувати зриви термінів запуску замовлень
	Врахувати пріоритет дорогих замовлень
	Врахувати пріоритети замовлень
Робочі центри	Максимізувати утилізацію робочих центрів
	Врахувати пріоритети завантаження вузьких місць
	Мінімізувати час простоїв РЦ
Комплексні	Мінімізувати простої, налагодження, пролежування (Мається на увазі («час наладки + час простою») * К - «час пролежування» -> MIN, де К - настраюється коефіцієнт, наприклад К = 1000)
	Мінімізація налагодження та пролежування

	Мінімізація налагодження, обробки та пролежування
--	---

У класичній багатокритеріальній оптимізації використовується принцип зважування критеріїв. Це означає, що кожному критерію необхідно поставити у відповідність його вагу і порахувати функцію оптимізації всіх критеріїв з урахуванням їх ваг. Наприклад, $f = c_1 \times y_1 + c_2 \times y_2 + \dots + c_n \times y_n$, де f - функція оптимізації, c_1 і y_1 - ваги критеріїв і їх значення.

У практичних завданнях зважити критерії і поставити їм у відповідність ваги часто досить проблематично. Це пов'язано, як правило, з відсутністю формалізованих правил і нормативно-довідкової інформації, що дозволяють це здійснити з достатньою точністю. Тому застосовуються принципи, які використовуються людиною при прийнятті рішень.

Крім цього, поставлені перед оптимізатором критерії суперечливі. Наприклад, «Мінімізація часу налагоджень» і «Виконання замовлень в строк». Тому, рішення завжди вибирається з наявних можливих варіантів, а кращим вважаються такі варіанти, які задовольняють поставленим вимогам. Тобто якщо встановити 80% важливості для критерію «Мінімізувати зриви термінів замовлень» і 60% важливості для критерію «Мінімізувати час налагоджень РЦ», то кращими будуть 20% непрострочених замовлень і 40% робіт, в яких час налагоджень мінімальний. Серед отриманого перетину і будуть знаходитися роботи, які задовольняють поставленим критеріям. Якщо критерії незалежні, то ймовірність знаходження роботи в перетині наступна: $p = 0,2 * 0,4 = 0,08$. Якщо залежні, то ймовірність збільшиться на коефіцієнт коваріації. Тобто, якщо маємо всього 100 робіт, то з великою ймовірністю 8 робіт будуть задовольняти поставленим вимогам. З цих 8 робіт необхідно вибирати одну - яка буде найкращим рішенням. Про засади вибору одного рішення з безлічі описано в реалізації алгоритму оптимізатора.

2.6. Пропонований алгоритм

В даній роботі обрано підхід який комбінує в собі жадібний алгоритм, який буде застосовано до горизонту плану в один місяць і метод математичної оптимізації, який буде застосовано до результатів отриманих жадібним алгоритмом на перші 1-5 днів розкладу.

Жадібний алгоритм з багатокритеріальною оптимізацією, заснований на евристичному методі оцінки призначень, дає прийнятний розв'язок з огляду на великий горизонт плану. Оптимальність отриманого розв'язку достатня для оцінки дат випуску замовлень, забезпечення матеріалами, завантаження обладнання. Необхідність планування на такий великий горизонт зумовлена великим циклом виготовлення замовлення.

Евристичні алгоритми ґрунтуються на емпірично перевірних рішеннях, які дозволяють досягти прийнятного результату.

Обраний метод відноситься до класу однократних алгоритмів складання розкладів званих диспетчеризації. Даний клас алгоритмів є достатнім у тому ж сенсі, що і весь клас однократних алгоритмів: кожен розклад можна отримати за допомогою деякої диспетчеризації. Характерним для цього класу є те, що послідовно призначаються моменти початку операцій для кожної машини, що утворюють строго зростаючу послідовність. Обраний метод полягає у формуванні правил або функцій переваги (пріоритетів). Для кожної i -ї операції з множини очікуючих на призначення, розраховується значення функції f_i уподобання і вибирається та операція, для якої f_i досягає максимуму або мінімуму. Множина операцій успадковує значення функції переваги роботи.

У даній роботі розглядається правило переваги, засноване на сортуванні робіт, а значить і порядку виконання операцій, за наступними правилами оптимізації.

В даних алгоритмах використовується область видимості операцій: перелік операцій, вхідні операції яких уже призначені.

Недоліки евристичних методів полягають в складності оцінки близькості отриманих розкладів до оптимального. Крім того, для кожної функції переваги існують завдання, для яких застосування даної функції призводить до результатів далеких від оптимального.

Для усунення цих недоліків вирішено використати ітераційний алгоритм математичної оптимізації, який неможливо застосувати для великого горизонту плану, але на малому горизонті він дозволяє значно покращити результати жадібного алгоритму. На вхід він отримує перші кілька днів розкладу складеного жадібним алгоритмом і в рамках уже обраного обладнання змінює порядок робіт для покращення обраних критеріїв оптимізації.

2.6.1. Головний цикл евристичного алгоритму пошуку початкового рішення

Крок 1. Початок.

Крок 2. Сформувати фронт робіт

Крок 3. Отримати для кожної роботи фронту робіт допустимі робочі центри, сформувати множину допустимих призначень: пар робота-робочий центр.

Крок 4. Обрати краще призначення з урахуванням обраної множини критеріїв та їх ваг (алгоритм обрання кращого призначення описаний у розділі 3.3.5).

Крок 5. Якщо залишились непризначені на робочий центр роботи, перейти на Крок 2.

Крок 6. Зберегти розраховані строки виконання робіт та їх призначення на обладнання у сховище.

Крок 7. Кінець.

2.6.2. Алгоритм вибору інтервалу призначення роботи на обладнання

Крок 1. Початок.

Крок 2. Отримати роботу для призначення на робочий центр.

Крок 3. Отримати ранній час початку роботи на робочому центрі з урахуванням переналагоджень і переміщень (алгоритм розрахунку раннього часу наведений у розділі 2.6.3).

Крок 4. Призначити роботу на ранній час початку виконання.

Крок 5. Кінець.

2.6.3. Алгоритм вибору інтервалу призначення роботи на обладнання

Під раннім часом початку роботи слід розуміти мінімальний час початку поточної операції з точки зору забезпеченості, тобто закінчення попередніх операцій. Цей час обчислюється щодо закінчення всіх попередніх операцій. Він також не може бути раніше за початок розподілу і фактичного часу початку, з урахуванням того, що алгоритм працює в реальному режимі часу.

Алгоритм розрахунку раннього часу початку роботи на робочому центрі наступний:

Крок 1. Початок

Крок 2. Отримати роботу і робочий центр, на який вона призначена.

Крок 3. Якщо робота фактично розпочата, то ранній час – це час фактичного початку, в іншому випадку перейти на Крок 4.

Крок 4. Вважати рекордом час початку розподілу. Перебрати роботи, що передують даній, якщо час закінчення операції більший за рекорд, то вважати час закінчення операції рекордом

Крок 5. Кінець.

2.6.4. Алгоритм вибору оптимального призначення по множині критеріїв

В даній роботі представлено два алгоритму багатокритеріальної оптимізації: позиційний та мінімаксний.

Позиційний алгоритм полягає у тому, що в порядку ваги критеріїв відбувається розрахунок значення оцінок й оцінка операцій. Такий підхід

дозволяє значно підвищити швидкість розрахунку так як не буде виконуватись оцінка по всіх критеріях відразу для всіх робіт, а отже значно підвищити швидкодію розрахунку.

Мінімаксний алгоритм полягає у тому, що розрахунок всіх часткових критеріїв виконується для кожної роботи, а отже кожен критерій завжди впливає на розрахований план.

2.6.5. Позиційний алгоритм оптимізації

Крок 1. Початок.

Крок 2. Отримати масив допустимих призначень: пари робота та робочий центр

Крок 3. Обрати краще призначення. Для кожного призначення виконати наступне:

- а) Розрахувати значення найвагомішого критерію.
- б) Отримати множину призначень, у яких значення критерію максимальне.
- в) Якщо у множині одне призначення – перейти до кроку 5.
- г) Інакше виконувати вибірку призначення з цієї множини, для розрахунку наступних по вазі критеріїв, до тих пір, поки не буде виявлено одне оптимальне призначення.

Крок 4. Виконати призначення виконання роботи на робочий центр.

Крок 5. Кінець.

2.6.6. Мінімаксний алгоритм оптимізації

У мінімаксному алгоритмі використовується наступні поняття:

Ліміт – це вага часткового критерію у процентах (від 0% до 100%)

Поріг – це кількість кращих призначень, що задовольняють частковому критерію. Значення порогу проходження призначення за частковим критерієм розраховується за наступною формулою:

$$t = \frac{\text{ліміт} \times \text{кількість призначень}}{100}$$

Мінімакський алгоритм виглядає наступним чином:

Крок 1. Початок.

Крок 2. Отримати масив допустимих призначень: пари робота та робочий центр

Крок 3. Розрахувати значення кожного часткового критерію для кожного призначення.

Крок 4. Для кожного часткового критерію розрахувати значення порогу і сформуванати масив кращих призначень.

Крок 5. Розрахувати кількість входжень призначень в множини кращих операцій по частковим критеріям.

Крок 6. Обрати краще призначення – те, що входить у більшу кількість множин кращих призначень по частковим критеріям.

Крок 7. Виконати призначення виконання роботи на робочий центр.

Крок 8. Якщо залишилися непризначені роботи - перейти до кроку 3.

Крок 9. Зберегти призначення операцій.

Крок 10. Кінець.

2.7. Алгоритм другого проходу

Розклад отриманий жадібним алгоритмом хоч і не є оптимальним але такий критерій, як рівномірне завантаження обладнання. Цього достатньо у випадках масового виробництва і виробництва на склад, а також виробництва де переналагодження не займають багато часу відносно часу виконання роботи.

У випадках коли виробництво отримує штучні замовлення які потрібно точно виконати до вказаної дати або у випадку наявності довго тривалих переналагоджень обладнання під різні типи продукції, жадібний алгоритм не буде давати оптимальні результати.

Для вирішення цієї задачі розроблено другий прохід алгоритму, який в рамках уже обраного обладнання і на основі уже отриманого початкового розкладу буде підбирати оптимальний порядок робіт. Застосувати цей алгоритм

для підбору обладнання недоцільно з точки зору часу розрахунку, отримана складність алгоритму не дозволить вирішити задачу за прийнятний час.

Алгоритм буде розроблено на основі принципу Constraint Programming(програмування обмеженнями). CP дозволяє описати задачу в вигляді декларативних обмежень, які будуть зведені до загального виду задачі. Загальною для всіх NP повних задач є задача SAT. Для задачі SAT уже існують відомі, описані в літературі і реалізовані в бібліотеках алгоритми, що дають допустимі рішення за прийнятний час.

Для застосування такого підходу необхідно описати задачу в вигляді цілочисельних змінних, множини обмежень, цільової функції та напрямку оптимізації(мінімізація чи максимізація). Розробка такої моделі для задачі календарного планування доволі складна, потребує попередніх розрахунків і глибокого розуміння предметної області.

На отриманій моделі необхідно запустити один із вирішувачів задач SAT. З огляду на кількість змінних і потужності множини обмежень необхідно застосовувати найсучасніший підхід до вирішення задачі SAT. Таким є алгоритмом CDCL, заснований на алгоритмі DPLL.

Головні кроки алгоритму:

1. Вибрати змінну та призначити їй True або False.
2. Збільшити номер рівня
3. Запам'ятати призначення в прив'язці до номеру рівня
4. Розповсюдження логічне обмеження (одиничне поширення)
5. Виявити отримані конфлікти призначень змінних, запам'ятати отримані конфлікти
6. Якщо конфлікти не виявлено – перейти на крок 1
7. Якщо конфлікт виправний то виконати зворотній прохід на номер рівня на якому виявили конфлікт
8. Якщо конфлікт не виправний то задача не може бути вирішена

Даний алгоритм реалізовано в бібліотеці з відкритим кодом Google OR-Tools, в вирішуванні CP-SAT. Відповідно необхідно розробити модель, яка дозволить звести задачу календарного плану до задачі, яка вирішується даною бібліотекою.

Ключовою відмінністю буде ітераційний підхід, на основі початкового плану алгоритм поступово буде знаходити кращі рішення. Це дозволяє у випадку вирішення об'ємної задачі, час знаходження оптимуму для якої надзвичайно великий, зупинити алгоритм на рішенні яке отримано за певний встановлений час, або на рішенні яке краще за початкове на певну величину.

Отриманий таким чином план хоч і не буде математично оптимальним, але з практичної точки зору він задовольнить потреби виробництва і уже дасть перевагу над конкурентами.

2.7.1. Цільова функція алгоритму другого проходу

На відміну від жадібного алгоритму на другому проході алгоритм дозволяє оцінити показники календарного плану в цілому, а не кожного окремого кроку. Відповідного до цього, для алгоритму розроблено комплексну цільову функцію, яка буде оцінювати оптимальність плану в цілому. Цільова функція має враховувати штрафи за зрив директивні термінів випуску замовлень, а також затрати які виробництво несе на випуск продукції. Задача алгоритму знайти розклад для якого ця цільова функція буде мінімально. Тому правильна ЦФ ключове при використанні запропонованого підходу.

Запропонована $ЦФ = Штраф \text{ за прострочку замовлень} + штраф \text{ переналагоджень} + штраф \text{ перележування незавершеного виробництва} + штраф \text{ відходів виробництва}$.

Далі проведемо пояснення кожної складової.

Штраф за прострочку замовлення = $\text{sum}(\text{кількість днів прострочки} * \text{ціна дня прострочки})$.

Штраф переналагоджень = $\text{sum}(\text{час переналагодження} * \text{ціну простою обладнання})$



Рисунок 2.1 – Приклад переналагодження

Для зміни розміру деталі яка виготовляється на обладнанні потрібно провести його переналадку, яка займає час. Переналадку зображено жовтим кольором. В цей час обладнання не може виконувати роботу і алгоритм повинен мінімізувати такий час за рахунок порядку робіт на обладнанні.

Штраф перележування незавершеного виробництва = $\text{загальний час між операціями над однією партією матеріалу} * \text{ціну партії матеріалу}$



Рисунок 2.2 – Приклад пролежування незавершеного виробництва

При такому запланованому виконанню першої синьої операції так, обладнання на якому потрібно виконувати наступну ще не звільниться до кінця першої. Таким чином буде спостерігатись явище "затарювання" - великих об'ємів незавершених замовлень, які не буде де зберігати. Крім того ці матеріали оптимальніше витратити на інше замовлення, яке необхідно виконати раніше або взагалі не замовляти. Штраф за такі рішення дозволить алгоритму побудувати більш оптимальний розклад.

Штраф відходів = $\text{sum}(\text{ціна матеріалу, які підуть у відходи})$

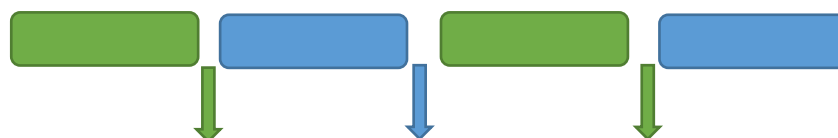


Рисунок 2.2 – Приклад відходів

Зміна кольору в якій лінія фарбує деталі веде до того, що багато фарби піде у відхід. Стрілка відображає момент в який відбувається зміна кольору. Таким чином, якщо алгоритм призначає роботи так, що постійно потрібно міняти колір, буде спостерігатись перерозхід матеріалів. Часто утворенню відходів сприяє переналадка. Накладання штрафів за порядок робіт, який веде до утворення відходів дозволяє отримати план, що дає конкурентні переваги в зв'язку з меншою собівартістю продукції.

2.7.2. Обмеження алгоритму другого проходу

В рамках концепції програмування обмеженнями на вхід алгоритму необхідно подати множину змінних і опис обмеження значень цих змінних. Алгоритм повинен виконати підбір множини значень змінних, які будуть задовольняти ці обмеження. Точність планування одна хвилина, тому алгоритм буде підбирати цілочисельні значення змінних де 1 це хвилина.

Деякі з обмежень можливо задати в вигляді лінійних рівнянь, інші зводяться до задачі SAT(задача здійсненності булевих формул).

Змінні для n операцій

- $i=0,n$
- $S(i)$ початок операції
- $F(i)$ кінець операції
- $D(i)$ тривалість операції

Константи

- S – початок розкладу
- F – кінець розкладу
- $DN(i)$ – нормативний час на виконання операції i

Базові обмеження

- $S \leq S(i) < F$

- $S \leq F(i) < F$
- $F(i) - S(i) = D(i)$
- $F(i) \geq S(j)$, де j наступна операція по технологічному маршруту за i
- $F(i) - S(i) = D(i)$
- $D(i) = DN(i) + \sum(Bd(j,e))$, де j , перерва у роботі для якої виконується умова $S(i) \leq Bs(j,e) \leq F(i)$. Призначення : описати ситуацію, коли робота почалась до перерви і продовжиться після.

2.7.2.1. Обмеження “час переналагодження”

Обмеження полягає в тому, що виконання наступної операції на обладнанні може потребувати переналагодження яке займає певний час. Час переналагоджень між операціями задано в вигляді матриці переналадок.

В загальному вигляді дане обмеження можна описати в вигляді рівняння $S(i) \geq F(j) + Cd(i,j)$, де i,j дві послідовні операції на одному обладнанні, $Cd(i,j)$ час переналадки між роботами i,j .

Опис цього обмеження в загальному вигляді не є оптимальним. Причиною цього є складність опису умови, того що дві операції послідовні. Умова послідовності операцій описана засобами ConstraintProgramming приводить до надвеликої розмірності задачі.

Більш ефективний спосіб описати обмеження це звести задачу переналагоджень кожного окремого обладнання до задачі TSP(комівояжер). Для цього необхідно сформулювати повнозв'язний, направлений, зважений граф в вершинах якого будуть операції. Ребра будуть відображати час переналагодження з операції i на операцію j . Додатково в граф необхідно ввести вершину яка буде відображати початок і кінець розкладу. На неї необхідно побудувати зв'язки з кожної роботи, і зворотні зв'язки. Вага зв'язків повинна дорівнювати нулю. Така вершина необхідна для забезпечення наявності циклу, що проходить через цю вершину і всі інші вершини.

Для опису задачі необхідно додатково ввести булеві змінні $C(j,j)$, що будуть показувати чи є робота i й j послідовними. Засоби CP дозволяють

описати умовні обмеження, які активуються в залежності від значення вказаної булевої змінної. Також CP передбачає таке обмеження як Circuit – цикл в графі.

В результаті обмеження буде виглядати так:

- Наявний один цикл в графі, який проходить тільки через активні ребра
- Ребро $i-j$ графа активно, за умови $C(j,j) = 1$
- Роботи i,j є послідовними за умови $C(j,j) = 1$
- Обмеження $S(i) \geq F(j) + Cd(i,j)$ діє за умови $C(j,j) = 1$

2.7.2.2. Обмеження “календарний час виконання операції”

Обмеження полягає в тому, що операція, яка розпочалась до перерви і закінчиться після перерви повинна календарно тривати сумарно нормативний час її виконання й час усіх перерв, що потрапляють в інтервал між початком і кінцем операції.

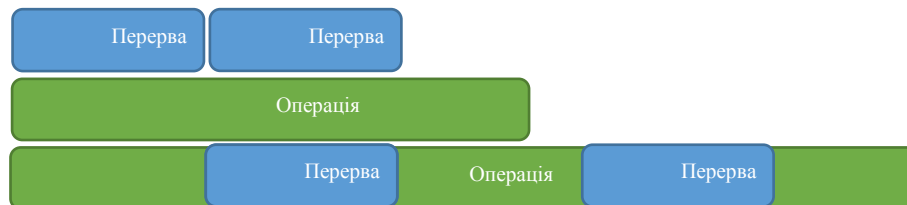


Рисунок 2.3– Приклад переходу через перерву

На рисунку зеленим кольором зображено операцію, а синім перерви в роботі обладнання. Таким чином операція витягується у часі при переході через перерви, і її тривалість буде рівна часу нормативному часу плюс час усіх перерв.

Розклад перерв в роботі обладнання задано в вигляді масиву часових інтервалів. Інтервал представляє собою неробочий час, вихідний, обідню перерву.

Для зручності будемо використовувати такі позначення

- **Bs(j,e)**, початок перерви j у роботі обладнання e
- **Bf(j,e)**, кінець перерви j у роботі обладнання e
- **Bd(j,e)**, тривалість перерви j

В загальному вигляді обмеження можна описати так $D(i) = DN(i) + \text{sum}(Bd(j,e))$, де j, перерва у роботі обладнання для якої виконується умова $S(i) \leq Bs(j,e) \leq F(i)$.

Дослідження складності отриманої задачі при такому способі задання обмеження показало, що отримана складність не дозволить вирішувати задачу за прийнятний час.

Для вирішення цієї проблеми необхідно змінити спосіб задання цього обмеження. Головна ідея це провести попередній обрахунок календарної тривалості операції для кожного можливого початку роботи. Тоді обмеження можна звести до умовного, що при потраплянні початку роботи в певний інтервал то тривалість її виконання буде вже відомою. Таке обмеження простіше і дозволяє вирішувати задачу за прийнятний час.

Розрахувати тривалість операції дозволяє застосування алгоритму заснованому на динамічному програмуванні. На початку алгоритму ми формуємо пустий список не завершених операції. Цикл алгоритму повинен проходити по кожній перерві і виконувати такі дії.

1. В список незавершених операції додати початки операції, що почались не раніше ніж за нормативний час операції до початку перерви
2. Пройти по списку незавершених операції, додати до їх часу виконання час перерви
3. Сформувати список незавершених операції з тих операцій, закінчення яких з врахуванням збільшеної тривалості потрапить на час після початку наступної перерви
4. Записати новий список незавершених в загальний список

Таким чином ми розказуємо результуючу тривалість операції для кожного можливого її початку.

Після отримання прив'язки тривалості операції до інтервалу початку дане обмеження можна описати в вигляді умовного обмеження

$$D(i) = Dc(I, s(i)), \text{ де } s(i) \text{ час початку операції.}$$

Висновки до розділу 2

При використанні позиційного алгоритму багатокритеріальної оптимізації, оцінка роботи за кожним частковим критерієм виконується у порядку ваги за потребою – у випадку, коли за більш вагомими частковими критеріями не вдалося однозначно отримати кращу роботу.

Перевагою цього алгоритму є те, що він дозволяє значно підвищити швидкодію розрахунку за рахунок відмови від безумовного розрахунку всіх часткових критеріїв для кожної роботи. Недоліком цього алгоритму є те, що менш важливі критерії не завжди впливають на план.

При використанні мінімаксного алгоритму багатокритеріальної оптимізації, для кожного можливого призначення роботи на робочий центр завжди виконується оцінка за всіма частковими критеріями, а кращим вважається те призначення, яке входить у найбільшу кількість множин кращих призначень за частковими критеріями.

Перевагою цього алгоритму є те, що для побудови плану завжди враховується кожен частковий критерій оптимізації. Недоліком цього алгоритму є недостатня швидкодія розрахунку на надто великих планах виробництва.

РОЗДІЛ 3

РОЗРОБКА СЕРВІСУ ПОБУДОВИ КАЛЕНДАРНОГО ПЛАНУ ЗАПРОПОНОВАНИМ АЛГОРИМОМ

3.1. Вимоги до програмного продукту

Задача побудови календарного плану є доволі ресурсозатратною. Промислові підприємства великої потужності потребують планування великого об'єму операцій, цей об'єм може сягати десятків тисяч операцій які необхідно розподілити на кілька сотень робочих центрів. Горизонт на який формується такий план може сягати від кількох тижнів до місяця. Розрахунок календарного плану зазвичай виконується кілька разів в день. Враховуючи такі об'єми і частоту розрахунку потрібно закласти можливість динамічного виділення ресурсів під кожен екземпляр задачі на основі її об'єму. Утримувати постійний час обчислювальний ресурс необхідний для роботи алгоритму – недоцільно. З огляду на описане вище головною вимогою буде використання хмарних рішень, з залученням необхідних потужностей лише на час роботи алгоритму.

Побудова оптимального розкладу не вирішує всі проблеми самостійно, а потребує тісної інтеграції з іншими рішеннями. Прикладами таких рішень є ERP системи, системи обліку, MES системи. Такі системи можуть бути розробленими різними постачальниками, працювати на різних операційних системах, розробленими на різних мовах програмування, розміщуватись як на серверах підприємства так і на потяжностях хмарних вендорів. Відповідно до таких обставин вимогою буде наявність WEB API. Зрозумілим і зручним для зовнішніх користувачів буде стиль REST API .

Так як розрахунки займають тривалий час і розрахунок календарного плану повинен запустити наступні етапи щоденної роботи підприємства, такі як видача завдань робочим, замовлення матеріалів, формування плану перевезення між обладнанням то необхідно мати засоби асинхронної комунікації для сповіщення й запуску обробки результату календарного планування. Для такого способу інтеграції доцільно використовувати протокол

AMQP, і реалізацію черги подій яка підтримує такий протокол. По результату розрахунку, в чергу обробки кожного підписаного сервісу має додаватись подія яка містить сформований календарний план. Таким чином інші системи зможуть оперативно реагувати на зміни календарного плану для забезпечення його виконання.

Існує велика кількість підприємств у яких немає автоматизованих програмних систем обліку замовлень, технологічної та конструкторської підготовки, але яким необхідно проводити розрахунок календарного плану. Також в багатьох випадках потрібно можливість провести демонстрацію результатів роботи алгоритму не інтегруючись з іншими системами. Для вирішення такої задачі необхідно надати можливість ручного завантаження вхідних даних у зручному вигляді. Звичним для таких випадків буде формат xls. Користувач повинен мати можливість імпортувати вхідні дані в форматі xls спираючись на попередньо підготований шаблон, та запустити розрахунок на завантажених даних.

Важливою частиною в календарному плануванні є можливість візуальної оцінки отриманого календарного плану. Стандартним засобом для візуалізації календарних планів є диграма Ганта. Так як алгоритм є ітераційним існує необхідність перегляду отриманого результату для кожної ітерації з можливістю порівняння й фіксації результатів. Крім того необхідно надати можливість вистежування зміни значення цільової функції з кожною ітерацією. Для кращого розуміння цього значення необхідно відобразити динаміку зміни її складових частин та інших KPI.

В деяких випадках, користувач може потребувати внесення ручних змін в план, такі зміни повинні розцінюватись як нова ітерація алгоритму. Для вирішення цих завдань необхідно розробити UI, у вигляді WEB застосунку, у якому буде реалізовано перегляд результатів розрахунку у вигляді діаграми Ганта, відображення графіків KPI. Діаграма Ганта повинна бути інтерактивною, дозволяти зміну порядку операції, переміщення на інше обладнання, при цьому

повинна виконуватись перевірка обмежень встановлених моделлю задачі і недопустимі дії мають блокуватись з відповідним інформуванням користувача.

Особливістю даної задачі є складка модель даних. Кожен екземпляр задачі являє собою одне ціле і містить великий об'єм інформації з складними зв'язками. Реляційна модель даних в такому випадку потребуватиме створення десятка таблиць, а запит на відбір міститиме велику кількість join. Тому для вирішення такої проблеми доцільно застосувати NoSQL сховище MongoDB, в якому кожен екземпляр задачі буде зберігатись в вигляді документу.

3.2. Структура програмного продукту

Враховуючи сформовані вимоги можна виконати проектування структури програмного продукту. На основі вимог вимальовуються такі складові частини:

1. WEB сервіс, який приймає заявки на розрахунок календарного плану, видає отримані результати.
2. Фонова служба, яка виконує розрахунок календарного плану певного екземпляру задачі.
3. WEB застосунок для аналізу отриманих результатів
4. Сховище результатів розрахунку та вхідних даних
5. Шина подій

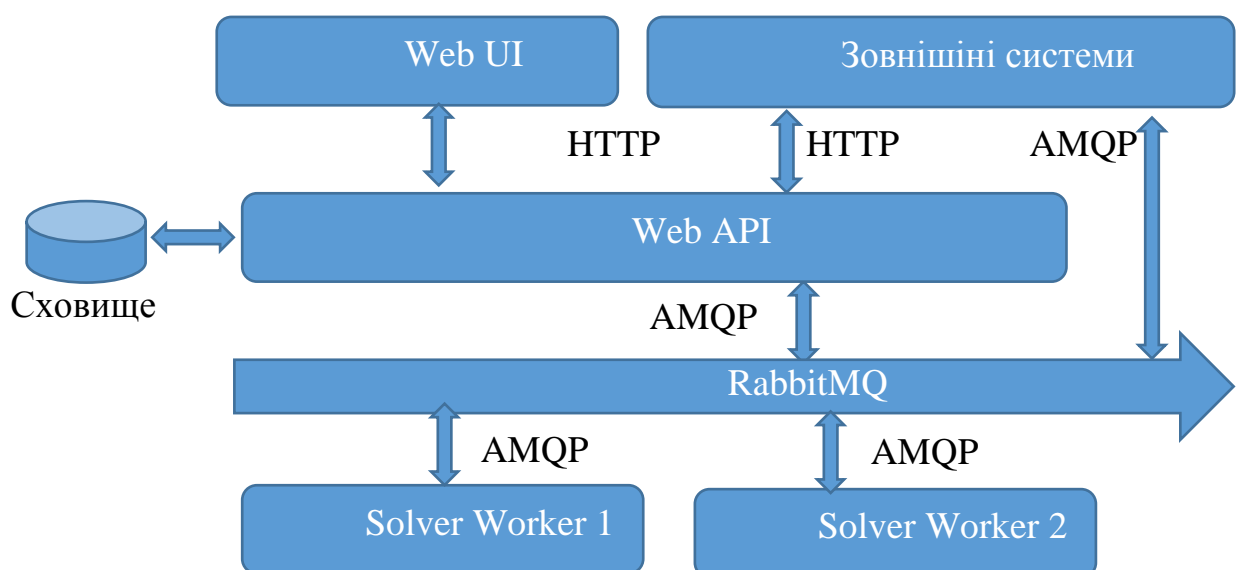


Рисунок 3.1 – Схема сервісу розрахунку календарних планів

3.3. Розробка WEB сервісу

Для реалізації сервісу обрано мову програмування C#, і платформу ASP.NET CORE 3.1. Обрана платформа дозволить реалізувати WEB API для запитів на розрахунок плану, отримання результату. Для online оновлення відображення на сторінці буде застосовано технологію SignalR, яка дозволяє викликати обробку серверної події на клієнті, по моделі підписки, без pooling запитів.

```

JobShopProblem {
  id integer($int64)
  timeoutInMinutes integer($int64)
  problemType ProblemTypes integer($int32)
  Enum:
    > Array [ 1 ]
  scheduleStart string($date-time)
  unlimitedLoadMachines {
    uniqueItems: true
    nullable: true
    string]
  breaks {
    > {...}
    nullable: true
  }
  changeovers {
    > {...}
    nullable: true
  }
  transfers > [...]
  orders > [...]
  machines > [...]
  machineSettings {
    > {...}
    nullable: true
  }
  operations > [...]
  operationsRelationns {
    > {...}
    nullable: true
  }
  operationsToMachine {
    > {...}
    nullable: true
  }
  horizonAtMinutes integer($int64)
  objective Objective integer($int32)
  Enum:
    > Array [ 9 ]
}

```

Рисунок 3.2 – Open API схема вхідних даних сервісу

```

MachineSchedule {
  operations {
    nullable: true
    readOnly: true
    OperationInterval {
      machine string
      start string($date-time)
      finish string($date-time)
      id integer($int64)
      type IntervalType
      Enum:
        > Array [ 4 ]
    }
  }
  changeovers {
    nullable: true
    readOnly: true
    ChangeoverInterval {
      machine string
      start string($date-time)
      finish string($date-time)
      type IntervalType
      Enum:
        > Array [ 4 ]
    }
  }
  breaks {
    nullable: true
    readOnly: true
    BreakInterval {
      machine string
      start string($date-time)
      finish string($date-time)
      type IntervalType
      Enum:
        > Array [ 4 ]
    }
  }
}

```

Рисунок 3.3 – Open API схема результату

Problems		▼
GET	/api/v{version}/Problems	
POST	/api/v{version}/Problems	
GET	/api/v{version}/Problems/{id}	
DELETE	/api/v{version}/Problems/{id}	
SolverJobs		▼
POST	/api/v{version}/SolverJobs	
GET	/api/v{version}/SolverJobs	
DELETE	/api/v{version}/SolverJobs/{jobId}	
GET	/api/v{version}/SolverJobs/{jobId}	
POST	/api/v{version}/SolverJobs/{jobId}/Cancel	
GET	/api/v{version}/SolverJobs/{jobId}/Status	
GET	/api/v{version}/SolverJobs/{jobId}/Result	
GET	/api/v{version}/SolverJobs/{jobId}/Schedule	

Рисунок 3.4 – Open API схема роботи з WEB сервісом

3.4. Розробка інтерфейсу користувача

Інтерфейс користувача повинен забезпечувати доступ до аналізу, коригування результатів розрахунку з будь якого пристрою, без виконання завантаження програмного забезпечення на клієнтську машину. Найкращий спосіб для досягнення цих цілей це реалізація інтерфейсу користувача в вигляді WEB-застосунку. Для WEB існує велика кількість готових компонентів для зображення діаграми Ганта та побудови графіків, які доцільно використати при розробці.

Ключовим елементом інтерфейсу користувача є діаграма Ганта. Елемент повинен дозволяти переглянути розподіл роботи по обладнанню, час на який призначена робота, зв'язки з іншими роботами, планові перерви роботи обладнання, планові переналагодження між роботами.

Відповідно до цих вимог розроблено структуру сторінки. Сторінка перегляду результатів розрахунку плану повинна мати 4 закладки:

- табличне відображення робіт;
- діаграма Ганта з горизонтальною віссю по часу;

- налаштування ваги критеріїв оптимізації;
- значення КРІ кожної ітерації розрахунку;

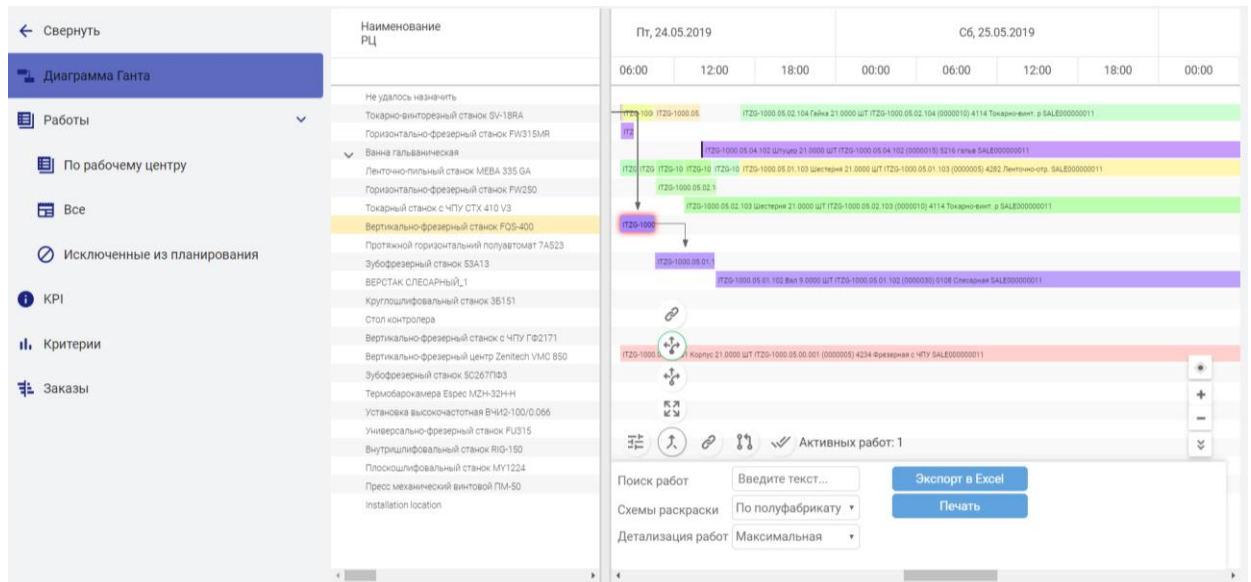


Рисунок 3.5 – Сторінка результатів розрахунку календарного плану

3.5. Оцінка розкладу отриманого за допомогою алгоритму

Аналіз результатів розрахунку календарного плану потребує глибоко розуміння предметної області та особливостей підприємства для кого виконувався розрахунок. З точки зору математики єдиним ефективним способом є оцінка значення цільової функції, але на практиці для людини складно співставити значення цільової функції, яка складається з багатьох складових й реальну якість отриманого плану. Для вирішення цієї задачі необхідно ввести показники на основі яких залежить цільова функція, або які дотично від неї залежні і при тому будуть зрозумілими людині. Ці показники називаються КРІ. КРІ виражають реальні, зрозумілі бізнесу величини на основі яких буде прийматись рішення про якість календарного плану.

Для кожного підприємства КРІ можуть бути різними в залежності від його спеціалізації та типу виробництва.

Враховуючи досвід впровадження системи вдалося сформувати список KPI, які найчастіше застосовуються на практиці:

- Тривалість розкладу - показує час від початку першої операції розкладу до останньої
- Якість обслуговування (за кількістю замовлень) - відсоток, розрахований як відношення кількості замовлень які заплановані в строк до загальної кількості наявних замовлень;
- Якість обслуговування (за вартістю замовлень) - відсоток розрахований як відношення вартості замовлень які заплановані в строк до загальної вартості наявних замовлень;
- Якість обслуговування поточного місяця (за кількістю) - аналогічно попередньому, але оцінюючи замовлення з директивним терміном в поточному місяці;
- Якість обслуговування поточного місяця (за вартістю);
- кількість замовлень пізніше терміну;
- завантаження обладнання;
- утилізація обладнання;
- середній цикл проходження замовлення - відношення суми циклів проходження замовлень до кількості замовлень;
- максимальний цикл проходження замовлення.

За аналіз якості отриманого календарного плану на підприємствах зазвичай виділяється окремий співробітник планово-диспетчерського відділу.

На першому етапі аналізу виконується перевірка наявності в причин непризначених операцій. Операції які не вдалось призначити на реальне обладнання призначаються на одне віртуальне обладнання. Це дозволяє не порушувати планування всього підприємства при наявності всього одного замовлення з неповним описом нормативної інформації.

Для аналізу причин неможливості розподілу робіт, розроблено спеціальний засіб в інтерфейсі користувача. Сторінка відображає всі роботи, які

не вдалось призначити, причини такого рішення алгоритму, рекомендації для усунення цих причин. Користувач має можливість ввести необхідну інформацію і повторити розрахунок.

Группы неназначенных работ										
Продукция, п/ф	Техмаршрут	№ оп. п/п	Операц (код)	Технологическая операция	Основное оборудование	Потребн	Укс	Завед	Кол-т	Продукция, п/ф
						длите	РЦ	РЦ	раб	(код)
ITZG-1000.05.01.100 Вал в сборе	ITZG-1000.05.01.100	0000005	0418	Комплектование					1	PB2000000005000
ITZG-1000.05.01.100 Вал в сборе	ITZG-1000.05.01.100	0000010	0401	Перемещение					1	PB2000000005000
ITZG-1000.05.03.100 Вал в сборе	ITZG-1000.05.03.100	0000005	0418	Комплектование					1	PB200000000E000
ITZG-1000.05.03.100 Вал в сборе	ITZG-1000.05.03.100	0000010	0401	Перемещение					1	PB200000000E000
ITZG-1000.05.02.100 Вал в сборе	ITZG-1000.05.02.100	0000005	0418	Комплектование					1	PB2000000009000

Неназначенные работы группы										
Заказ	Заказы MPS-плана	Наименование ресурса	РЦ	ФРЦ	ФЦ	Рабочий центр	Альтернативные РЦ	Все в РЦ		
SALE0000000011	SALE0000000011									

Доступные рабочие центры										
Оборудование	Продол:	ФГ	Оборуд	Групп	оснс	Рабочий центр(код)	Подр			
Отсутствуют данные для отображения										

Конфликты и решения		
Конфликт назначения	Решение, принятое системой	Рекомендации
Не удалось рассчитать потребность в оборудовании	Работа назначена на виртуальный рабочий центр с длительностью 0	1. Необходимо завести нормы использования оборудования;

Рисунок 3.6 – Сторінка аналізу нерозподілених операцій

Наступним етапом є аналіз якості отриманого плану згідно запропонованим KPI. Так як алгоритм складається з двох проходів, другий з яких є ітераційним, то необхідно мати можливість порівняти значення KPI отриманих на різних ітераціях. Це дозволить користувачу обрати план який буде більш оптимальним з його, суб'єктивно точки зору. Такий підхід є важливим, так як неможливо врахувати в системі всі фактори і закласти специфічні особливості кожного підприємства. Отримані ітерації порівнюють на сторінці KPI.

	Версия 8	Версия 7	Версия 6	Версия 5
Длительность расписания (в сутках)	21,19	21,19	20,29	20,29
Дата начала расписания				
Дата окончания расписания	13.06.2019 11:32:00	13.06.2019 11:32:00	12.06.2019 14:01:00	12.06.2019 14:01:00
Количество заказов (утверждено)	0	0	0	0
Количество заказов (прогноз)	0	0	0	0
Количество заказов (план)	0	0	0	0
Количество заказов (факт)	0	0	0	0
Количество заказов (позже срока)	0	0	0	0
Количество заказов (в срок)	0	0	0	0
Количество заказов (в срок, с допустимым опозданием)	0	0	0	0
Количество заказов (с допустимым опозданием)	0	0	0	0
Уровень обслуживания по количеству (в срок)	0	0	0	0
Уровень обслуживания по кол-ву (в срок, с доп.опозданием)	0	0	0	0
Уровень обслуживания по количеству (с допустимым опозданием)	0	0	0	0
Уровень обслуживания по стоимости (в срок)	0	0	0	0
Уровень обслуживания по стоимости (в срок, с доп.опозданием)	0	0	0	0
Уровень обслуживания по стоимости (с допустимым опозданием)	0	0	0	0
Суммарное время опоздания выпуска заказов (в сутках)	19,48	19,48	18,58	18,58
Среднее время прохождения заказов (в сутках)	21,15	21,15	20,25	20,25
Максимальное время прохождения заказов (в сутках)	21,15	21,15	20,25	20,25
Минимальное время прохождения заказов (в сутках)	21,15	21,15	20,25	20,25
Утилизация РЦ с ограниченной загрузкой (в процентах)	45,17	42,83	43,1	45,48
Утилизация РЦ с ограниченной загрузкой (в процентах)	12,93	12,53	12,77	13,67
Загрузка (в процентах)	12,93	12,53	12,77	13,67
Время простоев РЦ с ограниченной загрузкой (в сутках)	34,36	34,92	32,11	32,99
Время простоев РЦ с ограниченной загрузкой (в сутках)	109,04	109,04	103,08	103,08
Потребность (в сутках)	60,28	58,4	57,01	61,03
Длительность переналадок (в сутках)	0	0	0	0
Количество запланированных работ	83	83	83	83

Рисунок 3.7 – Сторінка KPI

По отриманим значення KPI користувач може зрозуміти, які потрібно змінити вагу складових цільової функції другого проходу і критерії жадібного алгоритму. Для цього в інтерфейсі існує окрема сторінка

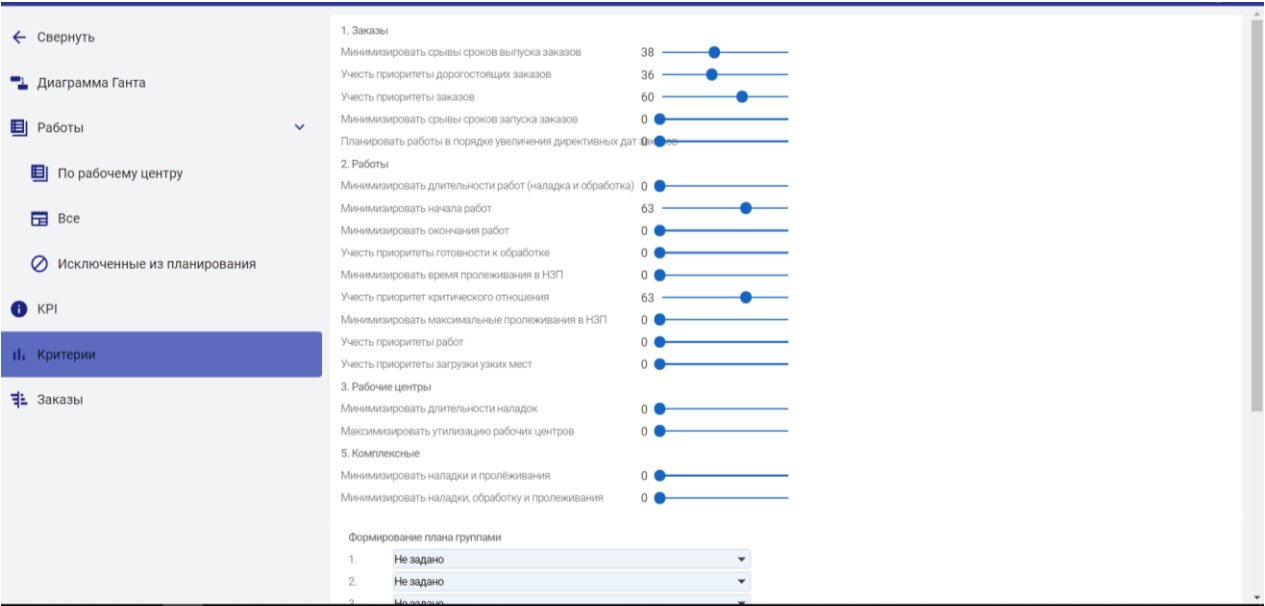


Рисунок 3.8 – Сторінка критеріїв оптимізації

Додатково існує можливість проаналізувати і порівняти завантаження різного обладнання.

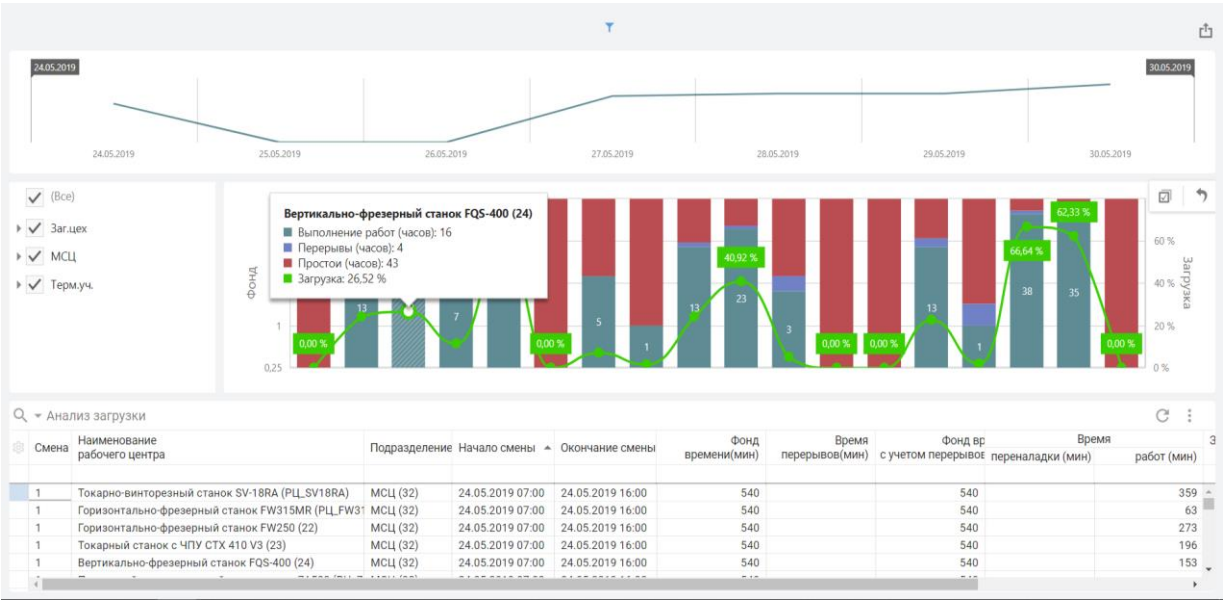


Рисунок 3.9 – Сторінка аналізу завантаження обладнання

Крім того важливим є можливість проаналізувати завантаження кожного окремого обладнання в різні періоди часу.

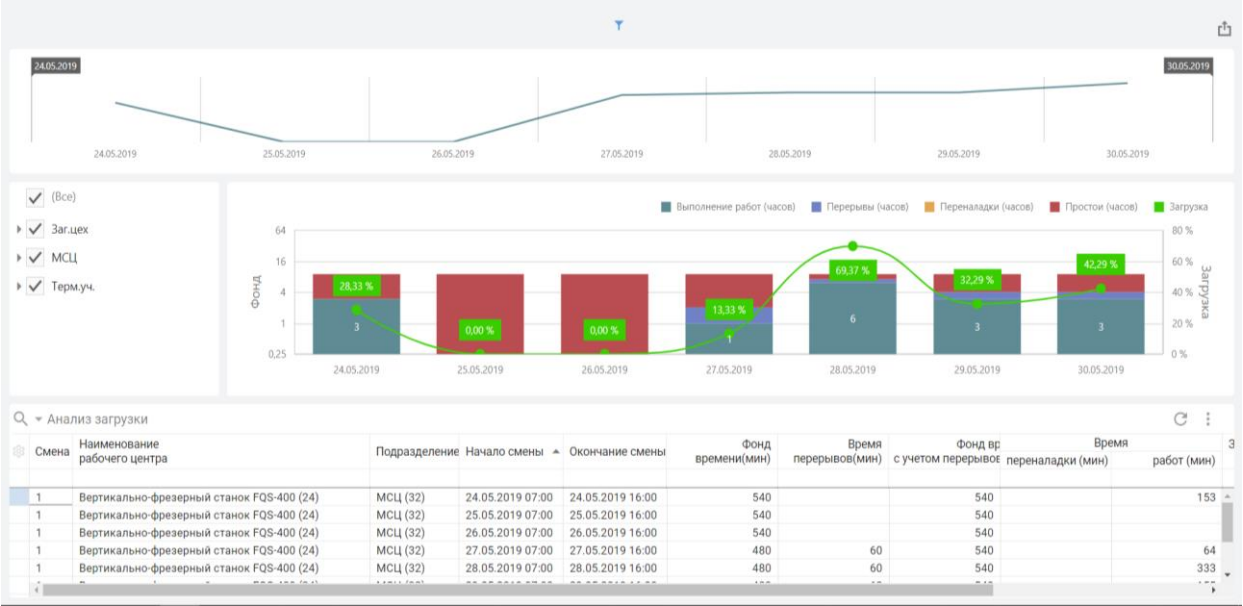


Рисунок 3.10 – Сторінка аналізу завантаження в періоди часу

3.6. Оцінка обчислювальних затрат на розрахунок плану

Розрахунок календарного плану на об’ємах реальних підприємств потребує колосальних ресурсів. Для оцінки потреби в ресурсах проведено

дослідження часу виконання алгоритму, завантаження CPU, та потреби в RAM для розмірностей задачі, що зазвичай зустрічаються на практиці.

Для проведення дослідження підготовано тестове середовище на машині розробника.

Таблиця 3.1 – Тестове середовище

Характеристика	Значення
Операційна система	Windows 10
Центральний процесор	Intel Core i7-8700k
Кількість ядер	6
Частота	3,7ГГц
Об'єм RAM	64 Гб

Масиви даних для проведення тестування підготовано способом розмноження одного і того ж набору замовлень і структури операцій для виявлення залежності саме від розмірності, а не структури.

Таблиця 3.2 – Результат досліджень затрат ресурсів

Показник				
Кількість операцій	3000	6000	150000	30000
Кількість одиниць обладнання	20	20	20	20
Споживання RAM	300 мб	1 Гб	7 Гб	34 Гб
Завантаження CPU	16%	16%	16%	16%
Тривалість розрахунку	7 секунд	40 секунд	2 хвилини	50 хвилин

Результат дослідження показує колосальні затрати оперативної пам'яті для виконання розрахунків великого об'єму, при малому завантаженні центрального процесору. Затрати такого об'єму пам'яті робить неможливим запуск кількох паралельних розрахунків на одній обчислювальній машині.

Крім того оцінка затрат пам'яті дає розуміння того, що WEB сервіс також має бути розміщено на окремій машині, для забезпечення прийому заявок на розрахунок плану з достатньою продуктивністю, незалежно від затрат ресурсів алгоритмом. Ріст потреби в пам'яті і часу розрахунку від розміру задачі показує, що необхідно забезпечити можливість припинення розрахунків розмірність задачі для яких не дозволяє отримати результат за прийнятний час, або веде до загальної перевитрати ресурсів, що в свою чергу буде заважати функціонування WEB сервісу і інших розрахунків.

Найкращим рішенням буде виділення WEB сервісу і алгоритму в окремі процеси, кожен з яких буде запущено в ізолюваному Docker контейнері. WEB сервіс має приймати заявку на розрахунок і запускати новий екземпляр алгоритму. Алгоритм повинен повернути результат в WEB сервіс через асинхронну чергу подій.

3.6. Напрямки подальших вдосконалень

Результати аналізу проведеному в попередньому розділі дозволяють зрозуміти, що при масовому використанні сервісу розрахунку календарних планів для масштабних задач, потреба в ресурсів буде колосальною. Постійне утримування такої кількості ресурсів приведе до надмірних затрат на інфраструктуру і високої собівартості кожного розрахунку. Крім того неможливого гарантувати що ресурсів завжди буде достатньо для обробки всіх запитів.

Для вирішення цієї проблеми доцільно скористатись послугами хмарних сервісів. Підходящим варіантом для реалізації алгоритму буде технологія Serverless. Для прикладу застосування технології AWS Lamda для реалізації

алгоритму дозволить динамічно виділяти необхідні ресурси на кожен запуск алгоритму. Що дозволить гарантувати обробку всіх запитів без затримки. Витрати при цьому будуть йти лише за необхідні ресурси, а не за простій чи покупку власного обладнання.

Висновки до розділу 3

В даному розділі виконано проектування, розробку та тестування сервісу, який реалізовує запропонований алгоритм вирішення задачі календарного планування. Розроблена архітектура дозволяє проводити ефективну інтеграцію з іншими системами які використовуються на підприємствах, проводити масштабування системи відповідно до кількості підприємств – користувачів.

Надані засоби аналізу і налаштування дозволяють перевірити правильність розрахунку й швидко налаштувати алгоритм під особливості конкретного екземпляра задачі. Засоби виявлення недоліків в вхідних даних дозволяють спростити користування сервісом і прискорити отримання користувачем результату розрахунку.

РОЗДІЛ 4

РОЗРОБКА СТАРТАП ПРОЕКТУ

4.1. Ідея проекту

В світі існує кілька десятків систем календарного планування виробництва. Особливістю цих систем є застаріла технологічна основа. Крім того зазвичай вони є невід'ємною частиною ERP системи. Впровадження цілої ERP системи є проектом який може тривати роками, що не дозволяє отримувати швидкий ефект від застосування календарного планування. Крім того вартість комплексної ERP системи є невід'ємною для малих і середніх підприємств.

Сукупність цих факторів веде до потреби в системі яка задовольняє такі критерії:

- Систему можливо швидко інтегрувати в уже існуючі рішення
- Система не потребує інфраструктурних затрат збоку підприємства
- Систему можливо налаштувати під особливості виробництва

4.2. Технологічний аудит ідеї проекту

Таблиця 4.1. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технологій
1	Хмарний сервіс розрахунку календарних планів	Мова програмування C#8	Є в наявності	Доступні безкоштовно
2		Платформа ASP.NET CORE 3.1	Є в наявності	Доступні безкоштовно
4		Розміщення на AWS	Є в наявності	Оплата по факту використання
3		Алгоритми розрахунку календарного плану	Є в наявності	Доступні безкоштовно

4.3. Аналіз ринкових можливостей запуску стартап-проекту

Для ефективного прийняття рішень по напрямкам розвитку продукту необхідно провести аналіз ринку. Це дасть розуміння хто є клієнтами, які цінності ми можемо їм надати. Також важливо розуміти конкуренцію в галузі, для впровадження рішень які дозволять надати краще рішення ніж конкуренти.

Таблиця 4.3. Аналіз конкуренції в галузі за М.Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Asprova, Siemens Preactor	MRPeasy, SAP	Локальні провайдери	Обмежені в виборі систем	Планування вручну
Висновки:	Конкуренти більше орієнтовані на великі підприємства чим на середні.	Орієнтовані на комплексні рішення	Постачання по моделі SAAS	Постачальники диктують цінові умови	Потрібна якість плану, яку людина не зможе надати

Таблиця 4.4. Характеристика потенційних клієнтів

п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Базова потреба, яку задовольняє товар (згідно концепції потенційного товару)	Визначити потенційні цільові групи клієнтів, що можуть бути зацікавлені у задоволенні означеної потреби	Вписати фактори, що формують поведінку клієнта (стандарти, технічні регламенти, інші фактори цінового та нецінового характеру) та особливості купівлі та експлуатації товару	- до продукції - до компанії-постачальника
	Збільшення ефективності виробництва шляхом роботи по оптимальному плану	Середні промислові підприємства	Розмір бюджету на ІТ рішення Відсутність кваліфікованих ІТ фахівців	Відсутність потреб до власної ІТ інфраструктури Ціна Оплата за результат Швидке впровадження

Таблиця 4.5. Фактори загроз

	Фактор	Зміст загрози	Можлива реакція компанії
--	--------	---------------	--------------------------

п/п			
	Низька якість календарного плану	Отриманий календарний план не дозволяє підняти ефективність виробництва	Залучення аналітиків, які сформулюють налаштування алгоритму, що ведуть до найкращих результатів
	Поява нових сервісів	Зниження кількості клієнтів	Створення доповнюючих сервісів, які гарно інтегруються з першим
	Криза в країні	Зниження платоспроможності клієнтів	Розробка особливі критерії оптимізації які дозволять краще працювати підприємству в умовах економії
	Низька кількість замовлень	Сервіс не набуде популярності у цільової групи	Кооперація з постачальниками суміжних рішень
	Витрати на утримання серверів перевищать надходження від доставок	Висока кількість розрахунків може призвести до надзвичайно великих витрат на обладнання	Використання хмарних сервісів для підключення обчислювальних потужностей лише за потреби

Таблиця 4.6. Фактори можливостей

п/п	Фактор	Зміст можливості	Можлива реакція компанії
	Велика кількість	За рахунок низької ціни	Реалізація можливості за рахунок збільшення

	підприємств - клієнтів	збільшується кількість підключених підприємств	кількості суміжних сервісів.
	Вихід на нові локації	Залучення користувачів в інших країнах	Відкриття центрів підтримки в інших країнах
	Зацікавленість постачальників комплексних рішень	Збільшення кількості клієнтів веде до зацікавленості відомих постачальників	Розширення засобів інтеграції, поставка сервісу в рамках інших систем
	Зниження цін на обчислювальні ресурси	Вартість утримування системи буде значно нижчою за доходи від комісії	Підключення можливості безкоштовного використання обмеженого набору функції для нових клієнтів

Таблиця 4.7. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
--------------------------------------	---	--

1. Вказати тип конкуренції - монополія/олігополія/ монополістична/чиста	олігополія	Інтеграція з наявними гравцями ринку. Збільшення якості алгоритму.
2. За рівнем конкурентної боротьби - локальний/національний/...	локальний	Детальне опрацювання особливостей кожного типу підприємств. Робота в типах виробництва не зайнятих конкурентами.
3. За галузевою ознакою - міжгалузева/ внутрішньогалузева	внутрішньогалузева	
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	Товарно - видова	Зарпопонувати новий рід товарів, що перетворить конкуренцію в товарно – родову.
5. За характером конкурентних переваг - цінова / нецінова	цінова	Використання інноваційних технологій для зниження витрат
6. За інтенсивністю - марочна/не марочна	Марочна	Вкладатись в впізнаваність бренду

Таблиця 4.8. Обґрунтування факторів конкурентоспроможності

п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
	Якість	Календарний план дозволяє

	календарного плану	підприємству працювати більш ефективно, таким чином отримувати більше вигоди на тих же виробничих потужностях.
	Ціна рішення	Інші постачальники орієнтуються на великі підприємства, що веде до надмірних цін продукт
	Відсутність потреби у власній IT інфраструктурі	Клієнти надають перевагу SAAS системам, що не потребують додаткових затрат на ресурси і адміністрування
	Стандартні протоколи інтеграції	Клієнти зазвичай уже мають певні системи і нове рішення потрібно з ними інтегрувати не замінюючи повністю.

Таблиця 4.9. Порівняльний аналіз сильних та слабких сторін

п/п	Якість календарного плану	Бал и 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1		1	2	3
	Якість календарного плану	18							
	Ціна рішення	10							
	Відсутність потреби у власній IT інфраструктурі	18							
	Стандартні протоколи інтеграції	20							

Таблиця 4.10. SWOT аналіз стартап-проекту

Сильні сторони: Орієнтація на використання сучасних технологій Пряма вигода для підприємств	Слабкі сторони: Новий для користувачів сервіс. Розрахунок на залучення середніх підприємств
--	--

Можливості: Сервіс буде видавати кращий календарний план ніж великі комплексні рішення	Загрози: Відсутність зацікавленості клієнтів Вхід технологічних гігантів на ринок систем планування
--	--

Таблиця 4.11. Альтернативи ринкового впровадження

п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
	Відмова від самостійного сервісу на користь інтеграції програмного продукту в існуючі комплексні системи.	Висока	6 місяців
	Відкриття алгоритму у вигляді відкритого програмного коду	Низька	відразу
	Реалізація безкоштовного сервісу з платними доповненнями	Середня	4 місяці

4.4. Розроблення ринкової стратегії проекту

Таблиця 4.11. Вибір цільових груп потенційних споживачів

п/п	Опис профілю цільової групи потенційних	Готовність споживачі в сприйнят	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
-----	---	---------------------------------	---	--------------------------------------	--------------------------

	клієнтів	и продукт			
	Малі промислові підприємства	Готові при швидкому впровадженні й низькій ціні	40% Продукт дозволить розраховувати календарний план не маючи власної інфраструктури	Низька. Більшість гравців орієнтуються на великий бізнес	Просто. Користувачі гарно сприймають нові продукти. Короткий ланцюжок прийняття рішень
	Середні промислові підприємства	Готові при реальних перевагах отриманого календарного плану.	30% Часто існуючі методи розрахунку плану не влаштовують але впровадження нових систем занадто трудоємке	Низька. Більшість гравців орієнтуються на великий бізнес	Просто. Потрібно довести, що рішення дозволяє працювати більш ефективно
	Великі промислові підприємства	Не готові. Проведення нових систем потребує погодження і проведення яке може тривати кілька років	5% Велика кількість розробляє власні системи.	Висока. Конкуренти SAP ASPRO VA PREAC TOR	Складно. Необхідно інтегруватись з комплексними рішеннями. Довести перевагу над власними розробками підприємства

Таблиця 4.12. Визначення базової стратегії розвитку

	Обрана	Стратегі	Ключові	Базова
--	--------	----------	---------	--------

п/п	альтернатива розвитку проекту	я охоплення ринку	конкурентоспроможні позиції відповідно до обраної альтернативи	стратегія розвитку*
	Безкоштовні пілотні запуски. Поширення результатів перших проектів. Відкритий доступ демонстрацій	Цільовий маркетинг	Підвищення ефективності підприємств змусить їх конкурентів по галузі до використання запропонованої системи	Лідер по витратах

Таблиця 4.13. Визначення базової стратегії розвитку

п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
	Так. Галузь управління виробництвом доволі відстала в плані технологій	Залучати нових	Ні.	Стратегія конкурентної ніші

Таблиця 4.14. Визначення стратегії позиціонування

	Вимоги	Ба	Ключові	Вибір асоціацій,
--	--------	----	---------	------------------

п/п	до товару цільової аудиторії	зона стратегія розвитку	конкурентоспроможні позиції власного стартап-проекту	які мають сформувати комплексну позицію власного проекту (три ключових)
	Ціна, швидкість впровадження, зручність в користуванні, відсутність супутніх затрат	Лідер по витратах	<ul style="list-style-type: none"> - Нижча ціна впровадження - Короткий термін оцінки ефективності 	<ul style="list-style-type: none"> -інноваційність -швидкий результат -поряма вигода

4.5. Розроблення маркетингової програми стартап-проекту

Таблиця 4.15. Визначення ключових переваг концепцій товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Дешеве рішення	Ціна обчислюється алгоритмом як мінімальна за яку може виконатись доставка.	Зниження власних витрат а рахунок використання хмарних сервісів і альтернатив можливо встановлювати ціну, що дає високу додану вартість
2	Швидкість впровадження	Мінімум необхідної вхідної інформації. Відсутність затрат часу на розгортання інфраструктури	Система взаємодіє лише з однією частиною підприємства що спрощує погодження впровадження рішення. Це дає можливість отримати швидкий результат.
3	Відсутність супутніх витрат	Власна інфраструктура.	Не потрібно утримувати власний ІТ відділ для розгортання і підтримки системи

Таблиця 4.16. Опис трьох рівнів моделі товару

Рівні	Сутність та складові
-------	----------------------

товару			
I. Товар за задумом	Система для організації розрахунку календарного плану виробництва. Дозволяє налаштувати критерії оптимізації, цільову функцію та обмеження під особливості підприємства-клієнта		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	
	1. Можливість обраховувати 100 планів одночасно секунду.	Нм	Тл
	2. Затримка на розрахунок плану на місяць не більше 8 годин.	Нм	Тл
	3. Максимальне відхилення від оптимуму цільової функції в результуючого плану не більше 10%	Нм	Тх
	Якість: система CI/CD. Автоматичні Unit i Integration тести.		
	Пакування: Система розгорнута на Amazon WebServices		
Марка: AiPS			
III. Товар із підкріпленням			
	Заготовані шаблони розрахунків для найбільш поширених типів виробництва.		
За рахунок чого потенційний товар буде захищено від копіювання: За рахунок прав на інтелектуальний ресурс.			

Таблиця 4.17. Визначення меж встановлення ціни

п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	300\$	100 000\$	2 000 000\$	500 - 5000\$

Таблиця 4.18. Формування системи збуту

п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптиміальна система збуту
	Клієнти виконують підписку на сервіс через сайт	Розміщення додатку в магазині розширень	0	Власний сайт

Таблиця 4.19. Формування системи збуту

п/п	Специфіка поведінки цільових клієнтів	Канал і комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Клієнт хоче отримати доступ до демонстрації безкоштовно	Інтернет.	Розміщення реклами на сайтах професійних виставок. Запис демонстраційних відео.	Впевнити користувача в працездатності алгоритму	“Перевір сам”
	Клієнт не хоче розгортати ІТ інфраструктуру	Інтернет.	Розміщення реклами на сайтах хмарних вендорів	Впевнити користувача що в нього не буде супутніх витрат	“Користуйся вже зараз”

Висновки до розділу 4

В четвертий розділі виконано розробку стартап-проекту, який дозволяє реалізувати потенціал запропонованого алгоритму. Для успішної реалізації

проекту проведено аналіз потреб користувачів, конкурентні можливості гравців ринку. Це дозволило знайти слабкі сторони існуючих пропозицій та розробити стратегію розвитку продукту яка дозволить отримати переваги над конкурентами.

Для успішної розробки стартап-проекту виконано наступні завдання:

1. Описано основну ідею проекту, цінності які він несе для споживача.
2. Вивчено потреби користувача, функціональні можливості продуктів конкурентів
3. Складено список переваг розроблювано продукту
4. Розроблено маркетингову програму, що дозволить ефективно просувати продукт

ВИСНОВКИ

Розроблена система збільшує оперативність і гнучкість виробничого планування, що дозволяє:

- набагато більш оперативно реагувати на зміну виробничої ситуації: на включення нових замовлень у виробництво, зміну пріоритетів виконання замовлень, вихід обладнання з ладу;
- точно прогнозувати терміни виконання виробничих замовлень.

У ході розробки комплексу завдань був розроблений евристичний алгоритм для формування APS-плану з урахуванням директивних термінів і пріоритетів виробництва продукції, який надає ефективне рішення за допустимий час.

Розроблено модель бази даних для обліку виробничої інформації, необхідної для APS -планування.

Для програмної реалізації була вибрана багаторівнева архітектура програмного забезпечення і створена об'єктна модель APS -планування. Розроблено посібник користувача, в якому описані основні варіанти використання комплексу завдань і проведено модульне тестування програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гаврилов Д.А. Управление производством на базе стандарта MRP, 2-е изд. — СПб.: Питер, 2005, 416 с.
2. Уоллас Т., Сталь Р. Планирование продаж и операций: Практическое руководство — СПб.: Питер, 2010, 272 с.
3. Stephen Toub (2010). Patterns of parallel programming. Microsoft Corporation. Web: <http://www.microsoft.com/en-us/download/details.aspx?id=19222>
4. Мауэргауз Ю.Е «Продвинутое» планирование и расписания (AP&S) в производстве и цепочках поставок / Москва: Экономика, 2012. — 574 с.
5. Питеркин С.В., Оладов Н.А., Исаев Д.В. Точно вовремя для России: Практика применения ERP-систем. — М.: Альпина, 2002. — 368 с.
6. Производственный менеджмент: Учебник /Под ред. В. А. Козловского. — М.:Инфра-М., 2003. — 574 с.
7. Эрик Эванс. Предметно-ориентированное проектирование (DDD): структура сложных программных систем — М.: ООО «И.Д. Вильямс», 2011. — 448 с.
8. Роберт С. Мартин. Быстрая разработка программного обеспечения: принципы, примеры, практика. — М.: ООО «И.Д. Вильямс», 2004. — 752 с.
9. Роберт С. Мартин. Чистый код: создание, анализ и рефакторинг. — СПб.: Питер, 2010. — 464 с.
10. Сімоненко В. П., Сімоненко А.В. Метод пошагового конструирования для составления расписания занятий в учебных заведениях. — К.: Наукові вістіНТУУ «КПІ», 2008 — 85 с.
11. Мартин Фаулер. Архитектура корпоративных программных приложений. — М.: ООО «И.Д. Вильямс», 2006. — 544 с.
12. Эндрю Троелсен. Язык программирования C# 2010 и платформа .NET 4.0. — М.: ООО «И.Д. Вильямс», 2011. — 1392 с.
13. Armin Biere, Marijn Heule, Hans van Maaren and Toby Walsch. Handbook of Satisfiability. — IOS Press, 2008. — 367 с.

14. Michael L. Pinedo. Scheduling Theory, Algorithms, and Systems. — Springer, 2008. — 528 с.
15. Michael L. Pinedo. Planning and Scheduling in Manufacturing and Services. — Springer, 2005. — 398 с.
16. Serhii Sirenko, International Book Series "Information Science and Computing").
[Режим доступу]: http://www.foibg.com/ibs_isc/ibs-07/IBS-07-p25.pdf
17. I. Lustig, Progress in Linear and Integer Programming and Emergence of Constraint Programming, — Foundations of Computer-Aided Operations (FOCAPO), — 2003, — 412 с.
18. L. Zeballos and G.P. Henning, A Constraint Programming Approach to the Multi-Stage Batch Scheduling Problem, — Foundations of Computer-Aided Operations (FOCAPO), — 2003, — 346с.
19. Stuart J. Russell and Peter Norvig, Artificial Intelligence A Modern Approach, — Prentice Hall, — 1995, — 713с.
20. Thomas H. CormenCharles E. LeisersonRonald L. Rivest Clifford SteinIntroduction to Algorithms , — The MIT Press , —2009, —213с.
21. Sedgewick, Robert; Flajolet, Philippe . An Introduction to the Analysis of Algorithms (2nd ed.) , — Addison-Wesley, — 2013, —423с.
22. Steven S. Skiena, The Algorithms Design Manual, — Springer, — 2020, — 821с.
23. Robert Sedgewick, Algoritms, — AdisonWesley, — 2011, —1021с.
24. Herrmann, Jeffrey W. Handbook of Production Scheduling, — Springer, — 2006, —421с.
25. Mauergauz, Yuri, Advanced Planning and Scheduling in Manufacturing and Supply Chains, —Springer, —2008, —431с.
26. Blazewicz, J., Ecker, K.H., Handbook on Scheduling, — Springer, — 2009
27. Ehrgott, Matthias, Multicriteria Optimization, — Springer, —2004, —331с.
28. T'Kindt, Vincent, Billaut, Jean-Charles Multicriteria Scheduling, — Springer, —2003, —231с.

29. Janota, Lynce. Theory and Applications of Satisfiability Testing – SAT, Springer —2009, —431c.
30. Mayoh, Brian, Tyugu, Enn, Penjam, Jaan (Eds.) Constraint Programming, Springer, —1994, —511c.
31. Apt K. R. Principles of Constraint Programming. New York: Cambridge University Press, —2003. —407c.
32. Rossi F., van Beek P., Walsh T. (eds.) Handbook Of Constraint Programming. Elsevier, —2006. —978 c.
33. Bessiere, Christian, Handbook of Constraint Programming, Foundations of Artificial Intelligence, Elsevier, —2003, —83c.
34. Cook, William J.; Cunningham, William H.; Pulleyblank, William R.; Schrijver, Alexander. Combinatorial Optimization. Wiley, —1997, —287c.
35. S. Martello, D. Pisinger, P. Toth, Dynamic programming and strong bounds for the 0-1 knapsack problem, Manag. Sci., —2009, —312c.
36. Gärtner, Bernd; Matoušek, Jiří. Understanding and Using Linear Programming. Berlin: Springer, —2006, —203c.
37. Lyle Ramshaw, Robert E. Tarjan .On minimum-cost assignments in unbalanced bipartite graphs. HP research labs, —2012, —67c.
38. Wenyu Sun; Ya-Xiang Yua. Optimization Theory and Methods: Nonlinear Programming, Springer, —2010, —541c.
39. H. Paul Williams. Logic and Integer Programming. Springer, —2009, —217
40. Donald E. Knuth, The Art of Computer Programming Volume 4, Fascicle 6: Satisfiability, —2012, —310c.
41. Martin Kleppmann, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems, OREILY, — 2017, —830c.
42. Titus Winters Software Engineering at Google: Lessons Learned from Programming Over Time, —2020, —1051c.
43. Mark Richards , Fundamentals of Software Architecture: An Engineering Approach, OREILLY, — 2020, —812c.

- 44. Maarten van Steen, Distributed Systems, CreateSpace, — 2019, —613c.
- 45. T. Walsh. SAT v CSP. In Principles and Practice, — 2011, —132c.
- 46. Lecture Notes in Computer Science, Springer, —2000, —456

ДОДАТОК 1

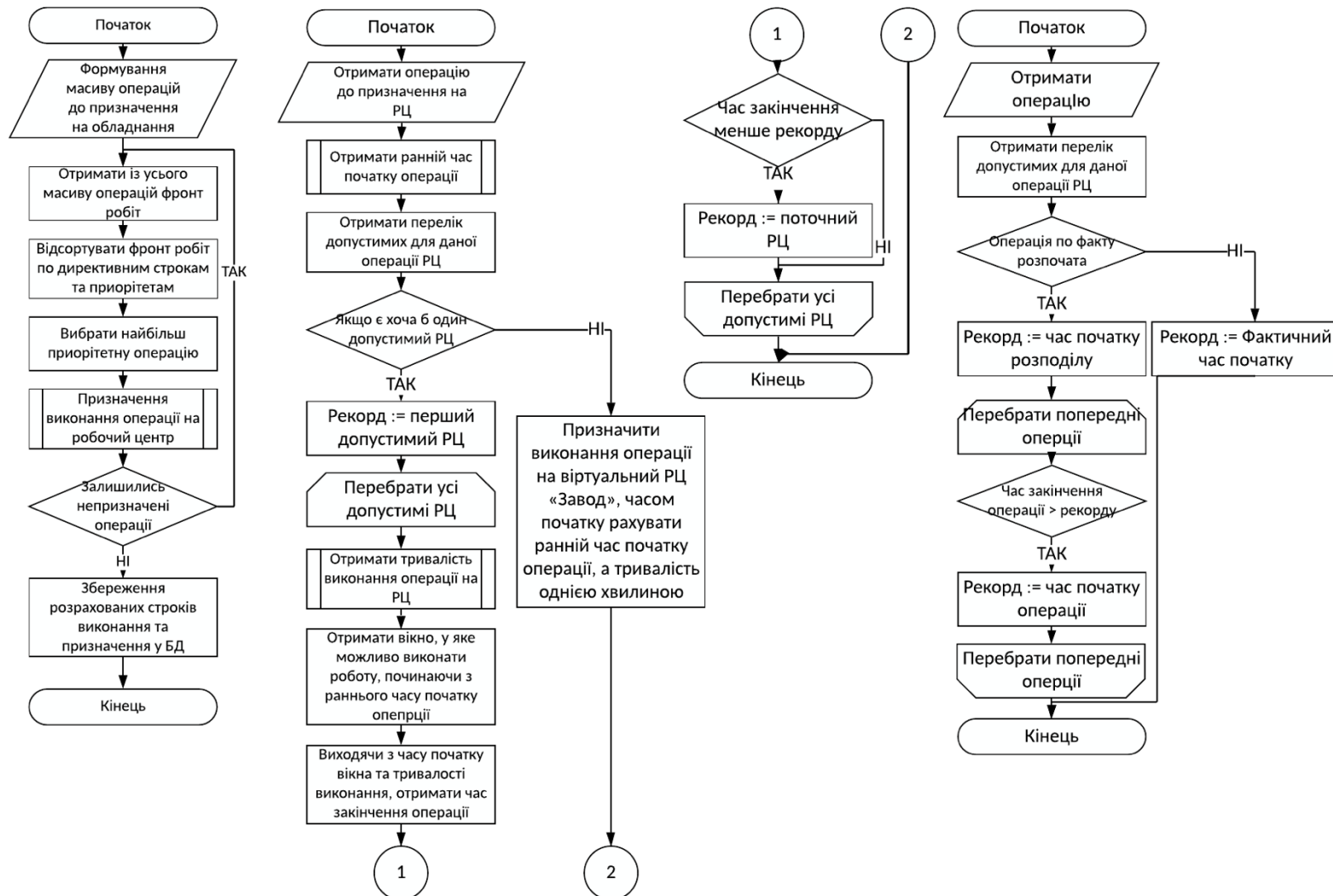
Метод багатокритеріальної оптимізації для задачі календарного
планування виробництва

Алгоритм призначення робіт на обладнання

Схема алгоритму

ІАЛЦ.467400.007 ДЗ

Аркушів 1



					ІАЛЦ.467200.007 ДЗ				
					Схема структурна алгоритму призначення робіт на обладнання	Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Гріщенко К.С.							
Перевірів		Сімоненко В.П.							
Т. кон.						Аркуш 1		Аркушів 1	
					Метод багатокритеріальної оптимізації для задачі календарного планування виробництва	КПІ ФІОТ Кафедра ОТ гр. ІО-391мп			
Н. кон.		Сімоненко В.П.							
Затвердив		Стіренко С.Г.							

ДОДАТОК 2

Метод багатокритеріальної оптимізації для задачі календарного
планування виробництва

Алгоритм розрахунку тривалості робіт

Схема алгоритму

ІАЛЦ.467400.007 ДЗ

Аркушів 1



					ІАЛЦ.467200.007 ДЗ				
					Схема структурна алгоритму розрахунку тривалості операції	Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Грищенко К.С.							
Перевірів		Сімоненко В.П.							
Т. кон.						Аркуш 1		Аркушів 1	
Н. кон.		Сімоненко В.П.			Метод багатокритеріальної оптимізації для задачі календарного планування виробництва	КПІ ФІОТ Кафедра ОТ гр. ІО-391мп			
Затвердив		Стіренко С.Г.							

